ÍNDICE

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

0 INTRODUCCIÓN

El presente proyecto se ha desarrollado en colaboración con *Gamesa Innovation and Technology* y dentro del Departamento de Ingeniería Matemática e Informática de la Universidad Pública de Navarra.

El proyecto versará sobre ensayos de fatiga en palas de aerogeneradores y más concretamente en simulación de los mismos.

El objeto del proyecto es desarrollar una aplicación en el entorno de Matlab que permita estimar:

- La frecuencia de ensayo.
- El número de actuadores a emplear y sus posiciones.
- El número de masas muertas y su colocación.
- La distribución de cargas lograda en función de los parámetros anteriores empleados.

El desarrollo de esta aplicación es algo necesario para la empresa, que en la actualidad no tiene ninguna herramienta que cumpla estas funciones. Lo que hacen actualmente es estimar previamente unos parámetros de partida en base a ensayos anteriores y pedirle al centro de ensayos que realice una simulación similar a la que se plantea en este proyecto.

Sin embargo la empresa preferiría realizar los cálculos previos al ensayo internamente. De este modo estarían seguros antes de enviar las cargas objetivo al centro de ensayos, de que estas son factibles con el número de masas muertas y actuadores seleccionados.

El proyecto se estructurará de la siguiente manera:

1. Introducción a la energía eólica: breve reseña histórica de la evolución de la energía eólica.
2. Fatiga: introducción de los fundamentos de la fatiga, fatiga en aerogeneradores, obtención de cargas y ensayos de fatiga en palas de aerogenerador.
3. Deducción de la ecuación armónica de ondas: descripción de la ecuación que modeliza nuestro problema.
4. Desarrollo del trabajo: planteamiento de los diferentes bloques del programa y lo que se hace en cada uno de ellos. Además, se exponen algunas cuestiones de implementación en Matlab.
5. Resolución de la ecuación armónica de ondas: desarrollo de los distintos métodos matemáticos empleados en la discretización de la ecuación armónica de ondas.
6. Análisis de resultados: Simulación y análisis de varios ensayos que nos permiten verificar el correcto funcionamiento del programa.
7. Conclusiones.
8. Bibliografía.
9. Anexos:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

a) Manual de usuario.

b) Código fuente.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

1 ENERGÍA EÓLICA

Hasta la aparición de la máquina de vapor en el siglo XIX, la única energía de origen no animal para la realización de trabajo mecánico era la proveniente del agua o del viento.

La primera y más inmediata forma de aprovechamiento de la energía eólica ha sido la empleada en la navegación. Las primeras referencias a embarcaciones a vela provienen de Egipto y datan del cuarto o quinto milenio antes de J.C.

Los molinos de viento llevan existiendo muchos años, si bien no está claramente definida la fecha en que se comienzan a utilizar.

Ya en la antigüedad. Persia, Irak, Egipto y China disponían de máquinas eólicas varios siglos antes de J.C. Hammurab I, rey de Babilonia, en el 1700 antes de J.C. utilizó molinos accionados por el viento para regar las llanuras de Mesopotamia y para la molienda del grano. Se trataba de primitivas máquinas eólicas de rotor vertical con varias palas de madera o caña, cuyo movimiento de rotación era comunicado directamente por el eje a las muelas del molino. Por su parte en China hay referencias de la existencia de molinos de rotor vertical y palas a base de telas colocadas sobre un armazón de madera, que eran utilizados para el bombeo de agua, máquinas conocidas como *panémonas*, precursoras de los molinos persas. Por último, el egipcio Hero de Alejandría representa en un estudio un molino de eje vertical de cuatro palas.

Los molinos de viento fueron utilizados en Europa en la Edad Media, comenzando a extenderse por Grecia, Italia y Francia. Si el origen de las máquinas eólicas presenta notables incertidumbres, no menos lo hace su expansión por el Mediterráneo y por toda Europa. Según algunos autores, la introducción de la tecnología eólica en Occidente se debe a los cruzados, si bien otros opinan que Europa desarrolla su propia tecnología, claramente distinta de la oriental, ya que en Europa se imponen fundamentalmente los molinos de eje horizontal, mientras que los molinos orientales eran de eje vertical.

Sea cual fuese la forma de aparición de estas máquinas en diversos países europeos, lo cierto es que se encuentran abundantes ejemplos de la importancia que los molinos de viento llegaron a tener en diversas aplicaciones; citemos como ejemplo relevante los literarios molinos castellanos utilizados para la molienda y los no menos conocidos molinos holandeses usados desde 1430 para la desecación de los *polders*, todos ellos de eje horizontal. En el siglo XVI Holanda perfecciona el diseño de los molinos y los utiliza para el drenaje; entre los años 1609 y 1612, Beemster Polder fue drenado con la ayuda de estas máquinas. Sin embargo, no sólo utilizaron los molinos para drenar el agua, sino también para extraer aceites de semillas, moler grano, etc. Precisamente el nombre de molinos proviene de este tipo de aplicaciones.

Una idea de la importancia que en el pasado adquirió la energía eólica nos la da el hecho de que en el siglo XVIII, los holandeses tenían instalados y en funcionamiento 20.000 molinos, que les proporcionaban una media de 20 kW cada uno, energía nada despreciable para las necesidades de aquella época.

La teoría de la aerodinámica se desarrolla durante las primeras décadas del siglo XX, permitiendo comprender la naturaleza y el comportamiento de las fuerzas que actúan alrededor de las palas de las turbinas. Los mismos científicos que la desarrollaron para usos aeronáuticos Joukowski, Drzewiechy y

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Sabinin en Rusia; Prandtl y Betz en Alemania; Constantin y Enfield en Francia, etc, establecen los criterios básicos que debían cumplir las nuevas generaciones de turbinas eólicas.

En los años 20 se empiezan a aplicar a los rotores eólicos los perfiles aerodinámicos que se habían diseñado para las alas y hélices de los aviones. En 1927, el holandés A.J. Dekker construye el primer rotor provisto de palas con sección aerodinámica, capaz de alcanzar velocidades en punta de pala cuatro o cinco veces superiores a la del viento incidente.

Betz demostró en su famoso artículo "Die Windmuhlen im lichte neverer Forschung" (Berlín 1927), que el rendimiento de las turbinas aumentaba con la velocidad de rotación y que, en cualquier caso, ningún sistema eólico podía superar el 60% de la energía contenida en el viento. Por lo tanto, los nuevos rotores debían funcionar con elevadas velocidades de rotación para conseguir rendimientos más elevados. La teoría demostró también que cuanto mayor era la velocidad de rotación menor importancia tenía el número de palas, por lo que las turbinas modernas podían incluso construirse con una sola pala sin que su rendimiento aerodinámico se viera disminuido significativamente.

El uso de los molinos como fuente de energía se remonta a finales del siglo XIX con el aerogenerador de 12 kW DC, desarrollado en Dinamarca y construido por Brush en EEUU. Sin embargo a lo largo del siglo XX hubo muy poco interés en el desarrollo de los aerogeneradores, salvo el de recargar las baterías de viviendas rurales aisladas, para que estás dispusieran de electricidad. Este interés desaparece con la expansión de la red eléctrica, al llegar ésta al entorno rural. Una excepción fue el aerogenerador de 1250 kW Smith-Putnam construido en 1941 en EEUU. Esta máquina tenía 53 m de rotor, control por *pitch* y con palas abatibles para poder reducir cargas. El control por Pitch es un sistema de control de las palas del aerogenerador para aumentar su eficiencia, que consiste en que la pala pueda girar sobre si misma, variando su posición respecto al viento. A pesar de que la viga de la pala rompió catastróficamente en 1945, siguió siendo el aerogenerador más grande construido durante unos 40 años.

Golding y Divone colaboraron fuertemente en el comienzo del desarrollo de los aerogeneradores. Ellos patentaron el aerogenerador de 30 m de rotor y 100 kW Balaclava en la USSR en 1931. También el Andrea Enfield de 100 kW y 24 m de diámetro en Reino Unido a comienzos del 1950. Esta última tenía palas huecas abiertas por la punta que captaban el aire, llevándolo a través de la torre donde otra turbina accionaba el generador. En Dinamarca se desarrollo en 1956 el aerogenerador Gedser de 200 kW y 24 m de diámetro, mientras que Electricote France en 1963 probo un aerogenerador de 1.1 MW con 35 metros de diámetro. En Alemania Hutter construyo en 1950 y 1960 varios aerogeneradores ligeros e innovadores. En la figura 1(a) se ve uno de los aerogeneradores de Hutter de dos palas que, como otras tecnologías alemanas, era muy avanzado para su época.

A pesar de los innumerables avances y del entusiasmo, entre otros, de Golding en la asociación de la investigación de la electricidad de Reino Unido, no hubo gran interés en el desarrollo de los aerogeneradores hasta que el precio del petróleo se disparo en 1973.

El repentino incremento del precio del petróleo provoco que los gobiernos sacaran subvenciones para programas de investigación y desarrollo. En EEUU esto llevó al desarrollo de una serie de prototipos, comenzando por Mod-0 de 38 m de diámetro y 100 kW en 1975 y culminando con la máquina Mod-5B de 97.5 m y 2.5 MW en 1987. Proyectos similares fueron llevados a cabo en el Reino Unido, Alemania y Suecia. Había dudas sobre qué diseño o arquitectura sería la más efectiva en cuanto a costes y se investigó con varios diseños innovadores a escala real. En Canadá un aerogenerador de eje vertical de 4 MW, Darrieus, fue construido e investigado en las instalaciones de ensayos de turbinas

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

de eje vertical de Sandia en EEUU. En Reino Unido se construyó alternativamente un aerogenerador de eje vertical de 500 kW. En la figura 1 (b) se muestra un aerogenerador de eje vertical.

Fig. 1 (a) : Aerogenerador diseñado por Hutter.

Fig. 1 (b) : Darrieus.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En 1981 se desarrollo y probó en EEUU un innovador aerogenerador de 3.5 MW. Durante un tiempo no se tuvo claro cuál era el número ideal de palas y mientras tanto se fueron desarrollando aerogeneradores de una, dos y tres palas.

El mundo de la ingeniería obtuvo gran cantidad de información de estos programas subvencionados por los gobiernos, funcionando los prototipos según diseño. Aún así, los grandes aerogeneradores daban problemas para determinados tipos de viento y su fiabilidad no era muy alta. Las grandes compañías comercializaban aerogeneradores de pequeñas dimensiones. En concreto, en California las ayudas económicas del gobierno resultaron en la instalación de un gran número de molinos de pequeñas dimensiones (<100kW). Estos modelos sufrieron diferentes problemas, pero al ser pequeñas sus dimensiones los problemas eran fáciles de reparar.

Fig. 1 (c) : Parque eólico con aerogeneradores de 1.5 MW.

El modelo de aerogenerador danés nació de un aerogenerador de tres palas con regulación por *stall* y velocidad de funcionamiento fija, con multiplicadora y acoplada a la máquina de inducción. La regulación por *stall* es una alternativa a la regulación por *pitch*, en la que en lugar de regular la pala haciéndola girar sobre si misma lo hace por pérdidas aerodinámicas. La velocidad de funcionamiento fija implica que las palas del aerogenerador giraban en todo momento a la misma velocidad. Esta arquitectura aparentemente muy sencilla ha probado su eficacia y se ha implementado en máquinas de

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

hasta 60 m de diámetro, alcanzando potencias de 1.5 MW. Los diseños comercializados actualmente han ido alcanzando las dimensiones de los grandes prototipos con los que se ensayaron en los años 80. Las investigaciones que entonces se llevaron a cabo en regulación total por *pitch*, velocidad de funcionamiento variable y nuevos materiales (composites), han sido aprovechadas por los diseñadores actuales. En la figura 1 (c) se muestran imágenes de un parque eólico de aerogeneradores actuales de 1.5 MW.

Como hemos indicado, el estímulo para el desarrollo de la energía eólica fue el gran y repentino encarecimiento del precio del petróleo en 1973 y la preocupación por saber que los recursos de combustibles fósiles eran limitados. Sin embargo, hoy en día la principal razón para el desarrollo de la energía eólica es que esta emite mucho menos CO_2 (a lo largo de todo su ciclo de vida) y que, por ende, puede ayudar en la lucha contra el cambio climático. En 1997 la unión europea acordó alcanzar a abastecer con energías renovables un 12 % de la demanda de energía para 2010. A partir de entonces se comprendió que la energía eólica iba a jugar un rol fundamental, incrementándose la potencia eólica instalada de 2.5 GW en 1995 a 40 GW en 2010. Algo que ha sido ampliamente superado, pues sólo en España la potencia eólica instalada es de alrededor de 20 GW. A nivel mundial, según la APPA (Asociación de productores de energías renovables), en la actualidad hay 200 GW eólicos instalados.

En la figura 1 (d) que se muestra a continuación aparecen los diez países con máxima potencia eólica instalada acumulada a fecha de diciembre de 2009. En ella se muestran las respectivas capacidades instaladas acumuladas en GW. España destaca por ser el cuarto país en potencia eólica instalada, con un 12.1 % de la potencia eólica instalada a nivel mundial. Sólo es superada por países como EEUU (22.3%), Alemania (16.3%) y China (15.9%).

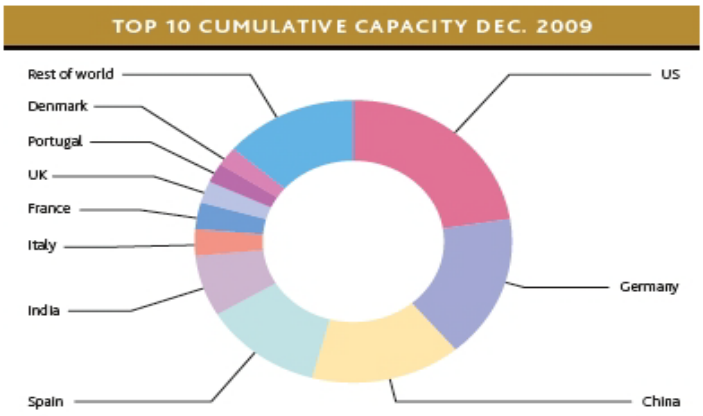
El sector eólico está en pasando un momento delicado actualmente debido al contexto de crisis internacional. Sin embargo, teniendo en cuenta el esperable crecimiento del precio del petróleo conforme sus reservas vayan bajando, las perspectivas son muy alentadoras. Además existe la certeza que el precio de la energía eólica va a bajar con el desarrollo de la tecnología.

En Europa los mejores emplazamientos eólicos se han ido agotando, por lo que el desarrollo de este sector se está dando en países emergentes como China y la India. Es en estos países donde las grandes empresas del sector se están intentando instaurar para mantener su crecimiento.

Para seguir instalando potencia eólica y poder lograr una reducción de emisiones de CO_2, se está comenzando con el desarrollo de la eólica *offshore*, que consiste en la instalación de aerogeneradores en el mar. Si bien, a priori, los costes son mucho más elevados, la esperanza de vida de un aerogenerador en el mar es del doble que la de uno terrestre, gracias a que los vientos son mucho más estables. El que los vientos sean más estables favorece a su vez la producción de energía eólica, si bien los costes de mantenimiento son más elevados.

Los aerogeneradores de la actualidad con rotores de entre 50 y 90 m, están evolucionando hacia aerogeneradores de dimensiones muy superiores. Hay empresas trabajando en rotores de entre 120 y 140 metros. Estos molinos serían ideales para su empleo en el *offshore*, si bien se prevé su utilización también en tierra.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira



	MW	%
US	35,159	22.3
Germany	25,777	16.3
China	25,104	15.9
Spain	19,149	12.1
India	10,926	6.9
Italy	4,850	3.1
France	4,492	2.8
UK	4,051	2.6
Portugal	3,535	2.2
Denmark	3,465	2.2
Total top 10	**136,508**	**86.5**
Rest of the world	**21,391**	**13.5**
World total	**157,899**	**100**

Fig. 1 (d) : Distribución de la potencia eólica instalada a fecha de Diciembre 2009.

Está evolución hacia aerogeneradores de mayor potencia y dimensiones, está exigiendo a una industria joven, como es la de la energía eólica, evolucionar a gran velocidad. El tener que competir con energías ya muy desarrolladas como las ligadas al petróleo o la energía nuclear, exige que existan subvenciones para que el precio de la energía eólica sea competitivo. Sin embargo, a diferencia de otras renovables, como la solar, el precio de la eólica es mucho más competitivo. Además, se espera que el precio del kWh eólico siga bajando a medida que el sector siga desarrollándose y se automaticen procesos tan manuales como el de la manufactura de las palas, proceso éste que podría considerarse más una obra de artesanía que de ingeniería.

Por tanto, queda claro que el futuro del sector dependerá fuertemente de que se consiguan reducir los costes del kWh eólico que la energía eólica sea competitiva sin la necesidad de subvenciones.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

2 FATIGA

2.1 INTRODUCCIÓN A FATIGA

En este apartado se va a explicar el fenómeno de la fatiga, dado que el programa desarrollado tiene como fin simular un ensayo de fatiga. Esta simulación nos permite conocer la distribución de esfuerzos (momentos), que sufrirá la pala durante el ensayo.

En tanto que en la actualidad todas las palas de los aerogeneradores se fabrican con materiales compuestos, la fatiga que estudiaremos será la relativa a los materiales compuestos.

En primer lugar, un material puede fallar bajo dos tipos de carga:

- Cargas estáticas.
- Cargas dinámicas.

El fallo por carga estática es aquel en el que le damos al material tiempo a deformarse. Bajo cargas de este tipo el material falla cuando alcanza su tensión máxima o su límite de deformación. Estos límites suelen estar muy estudiados y este tipo de fallos es fácilmente reproducible en ensayos estáticos.

Sin embargo el 90% de los componentes estructurales fallan por cargas dinámicas, lo que se conoce como fatiga. Este fallo tiene lugar sin que el material alcance su límite máximo de tensión o deformación. Tras un determinado número de ciclos dependiente de la carga, el material fallará sin que se haya alcanzado el límite elástico del material.

2.1.1 DIAGRAMAS *S-N*

La carga de fatiga es generalmente representada por una forma de onda cíclica, generalmente sinusoidal. Las características de esta onda se muestran en la figura 2 (a) izquierda, en términos de las magnitudes *S* y *N*. Como *S* se puede tomar tensión máxima o deformación máxima. Generalmente se usa tensión para los materiales homogéneos y deformación para los *composites* (son materiales anisótropos). La variable *N* es el número de ciclos a los que falla la pieza para una determinada amplitud de carga. Por lo tanto, cada punto de la curva *S-N* lo que describe es el número de ciclos hasta el fallo (la vida) que podrá soportar la pieza a un nivel de cargas con amplitud máxima S.

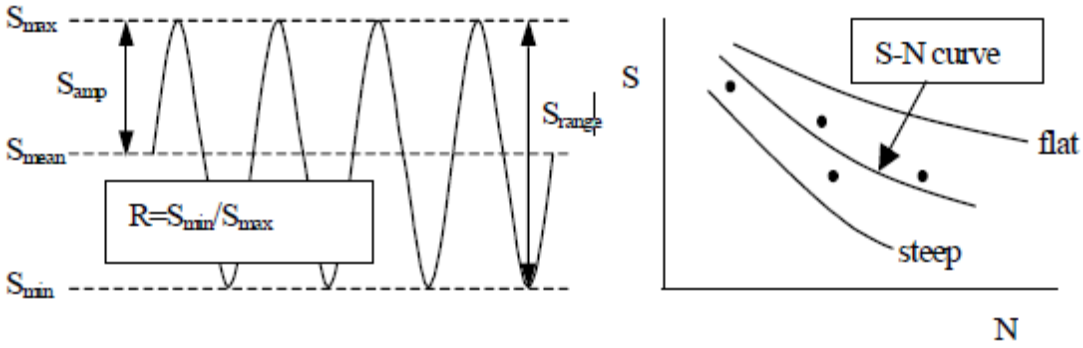


Fig. 2 (a): Terminología general de fatiga.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Tener en cuenta que *N (*número de ciclos) es la variable dependiente, que sin embargo se coloca en el eje de abcisas por tradición. Además es frecuente que en el diagrama *S-N* se considere escala logarítmica para el eje N, lo que nos permite observar la curva *S-N* como una poligonal.

Los diagramas *S-N* se obtienen de manera experimental y son muy costosos de obtener. El procedimiento a seguir consiste en ensayar hasta rotura especímenes del mismo material y con la misma geometría para diferentes niveles de carga. En el caso de palas de aerogenerador el precio de una de estas es elevadísimo, por lo que no es factible que para cada modelo se hagan todos los ensayos necesarios para describir el diagrama *S-N*. Estos diagramas se pueden acordar con las entidades certificadoras en base a la experiencia

Se comienza ensayando especímenes a un nivel con una amplitud de carga ligeramente inferior al límite estático de la pieza. Luego se van ensayando otros especímenes a un nivel de carga inferior al anterior y así sucesivamente recorriendo todo el espectro de amplitudes de carga entre el límite estático y la amplitud casi nula. Para cada nivel de carga se ensayan varios especímenes de manera que después de recorrer todos los niveles de carga tenemos una nube de puntos.

A partir de la nube de puntos experimental y mediante diversos métodos estadísticos se realiza un ajuste de la curva *S-N* que tiene la siguiente forma:

$$\log N = a + d \cdot S$$

donde:

- S es la amplitud de tensión o deformación máxima del material.
- N es el número de ciclos antes de fallo con una amplitud de carga S.
- a y d son parámetros dependientes de la naturaleza del material, de la geometría del espécimen y del tipo de solicitación.

Para obtener los parámetros a y b de la curva *S-N*, se utilizan distintos software que estiman estos parámetros ajustando las curvas por diversos métodos, siendo el más utilizado de todos ellos el de regresión lineal. Al final lo que hacen todos ellos es ajustar la curva, $\log N = a + d \cdot S$, a los datos experimentales. Estimando a y b, de manera que la curva se ajuste a la nube de puntos que forman el conjunto de los datos experimentales.

Cada curva *S-N* depende de infinidad de factores, teniendo una distinta para cada material, cada geometría y cada tipo de carga. Esto es algo que dificulta mucho los ensayos, dado que es imposible reproducir las condiciones exactas así como los ciclos en el caso de las palas de los aerogeneradores. Es decir, que para cada modelo de pala y cada solicitación necesitamos un diagrama.

Por otra parte, dado que según el tipo de carga que tengamos el diagrama *S-N* cambia, para acabar de describirlo necesitamos un parámetro, R, que de la relación entre la tensión o deformación máxima y la mínima.

$$R = \frac{S_{max}}{S_{min}}$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Para cada R tendremos una curva *S-N* distinta, como se mostrará gráficamente más adelante.

En el caso de palas de aerogenerador el precio de una de estas es elevadísimo, por lo que no es factible que para cada modelo se hagan todos los ensayos necesarios para describir el diagrama *S-N*. Estos diagramas se pueden acordar con las entidades certificadoras en base a la experiencia.

En algunos materiales como el acero hay una carga para la cual la curva *S-N* se hace horizontal, este es el límite de fatiga. Para cargas por debajo del límite a fatiga el componente no fallará por muchos ciclos de carga que sufra. Sin embargo en los materiales compuestos no existe el límite a fatiga, es decir, que por muy pequeñas que sean las cargas habrá un número de ciclos para el cual fallará el componente.

Los resultados de cálculos a fatiga están fuertemente sujetos a la formulación de fatiga que se escoja. Para subsanar los errores derivados de la formulación escogida, las distintas normativas te imponen unos coeficientes de seguridad a aplicar a las cargas de fatiga.

Los ensayos estáticos están muy estandarizados, mientras que para ensayos de fatiga de materiales compuestos hay una cantidad muy limitada de pautas para llevar a cabo los ensayos.

Cabe destacar que los materiales compuestos se comportan muy bien frente a fatiga, debido a la lo planas que son sus curvas *S-N*. Esto es así dado que cuanto menor sea la pendiente menor será la sensibilidad del material a un aumento del número de ciclos o de la carga. Además la dispersión en la vida a fatiga de los materiales compuestos es mucho mayor que para los metales.

Las curvas *S-N* para materiales compuestos se caracterizan por no tener límite a fatiga. Es decir que por muy bajas que sean las cargas habrá un número de ciclos para el cual el material fallará.

La flexibilidad en el diseño de los laminados es el punto fuerte de los materiales compuestos. La variación de las direcciones de fibra, el tipo, el contenido, la matriz… permiten diversos diseños que adaptables a cada aplicación concreta. No obstante, son muy difíciles de caracterizar en cuanto a fatiga. Hay que tener en cuenta que no hay dos estructuras de materiales compuestos iguales. Están sujetos a mucha incertidumbre: orientación de la fibra, adhesión fibra-matriz, presencia de deformaciones residuales…

2.1.2 PREDICCIÓN DE VIDA A FATIGA

La predicción de vida para cargas de fatiga de amplitud constante o variable para cualquier material pueden clasificarse en cuatro tipos:

- Daño acumulado.
- Daño progresivo.
- Fenómenos empíricos.
- Micro-mecánica.

En nuestro caso vamos a analizar la fatiga desde el punto de vista del daño acumulado. Seguimos este método porque así lo contemplan normativas como Det Norske Veritas (DNV, ver [17]) y Germanischer Lloyd (GL ver, [16]), siendo éstas dos de las más importantes entidades certificadoras.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El termino daño se usa para cualquier tipo de deterioro físico del material. La suma de Miner, explicada más adelante, describe la acumulación de daño de manera lineal sin especificar el tipo de daño de que se trata. En la práctica el daño está relacionado con la aplicación concreta de la que se trate. Además el daño puede expresarse en términos de degradación de la rigidez o de probabilidad de fallo. Para diseños basados en la fiabilidad, una probabilidad de fallo puede ser empleada como definición de fallo.

El método más comúnmente utilizado que nos permite medir el daño acumulado es la suma de Miner. Éste método no desprecia el daño causado por ningún ciclo. Con esta metodología el orden en que se producen los ciclos no se tiene en cuenta para predecir la vida a fatiga. El parámetro de daño es adimensional y el fallo se supone cuando el daño alcanza el valor de 1.

Una manera de ver el daño o la degradación en los materiales compuestos es analizar la evolución de la rigidez en el material. Es una medida cuantificable de manera no destructiva de conocer la resistencia a fatiga de un material. A medida que el material compuesto se va degradando, la rigidez del mismo va disminuyendo. Por tanto, a partir de la medida de la evolución de la rigidez podremos estimar la resistencia a fatiga del mismo y, lo que es más importante, la vida que le queda al material o la pieza en cuestión.

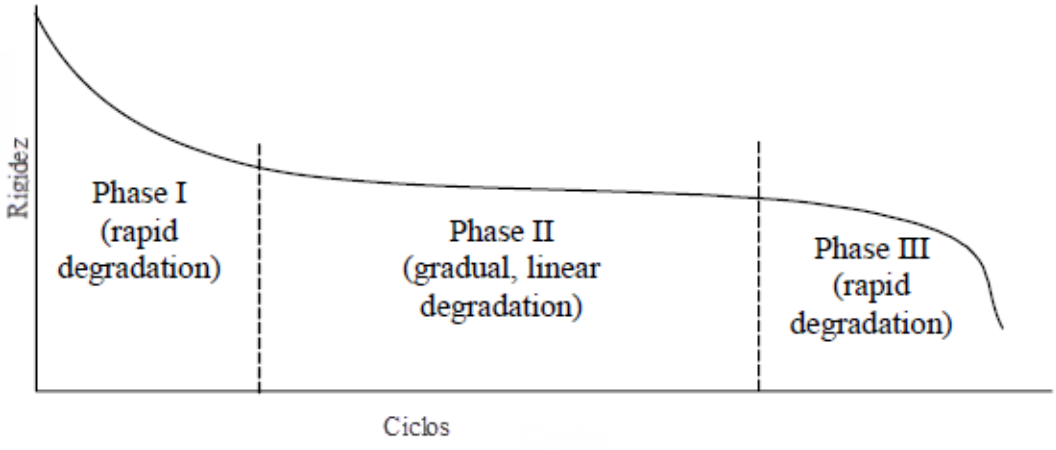


Fig. 2 (b): Degradación típica de la rigidez en los materiales compuestos.

En la figura se observan tres fases claramente definidas:

- Fase 1: se da durante los primeros ciclos, en los que la rigidez del material disminuye rápidamente. Por lo tanto, el material se degrada a gran velocidad.
- Fase 2: es la que ocupa la mayor parte de la vida del material. En ella el material se degrada a baja velocidad y de manera lineal.
- Fase 3: se da al final de la vida del material cuando, nuevamente, se acelera la degradación del mismo.

La degradación de la rigidez del material va a depender de las propiedades del material así como del tipo de cargas de fatiga.

2.1.3 AMPLITUD DE CARGA CONSTANTE

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Las cargas de fatiga en la vida real son de naturaleza muy compleja. Es evidente que los ciclos de carga son muy variados y no tenemos una carga oscilatoria repitiéndose de manera constante. Es por ello que, como se explicará más adelante, se hace un recuento de ciclos de manera que agrupamos las cargas de fatiga por grupos de carga en los que coinciden la carga media y la alterna. Es decir, en cada uno de esos grupos sí que tendremos una carga oscilatoria de la misma forma que se repetirá durante un determinado número de ciclos. Por lo tanto, en este estudio nos centraremos en cargas de amplitud constante y en el daño que estas producen.

2.1.3.1 CURVAS CLD

El diagrama CLD es una representación de los datos del diagrama *S-N*. Las líneas de vida constante de las curvas CLC conectan puntos que tienen la misma vida estimada, como una función de la tensión o deformación media y estimada. En la figura 2 (c) se puede ver un ejemplo. En la parte derecha de dicha figura está el diagrama *S-N*, del que se extraen las deformaciones máximas y a partir la R de cada una de estas se transforman esas deformaciones máximas en amplitud de deformación y media de deformación. A partir la amplitud de deformación y de la deformación media de cada uno de los dos casos obtenemos dos puntos para el diagrama CLD.

Los dos puntos del diagrama CLD se pueden unir entre sí por una recta. Por otro lado, el punto correspondiente a R > -1 (en este caso R=0.1), lo unimos al límite de tracción estático (UTS). De esta forma hemos creado una línea de vida estimada constante.

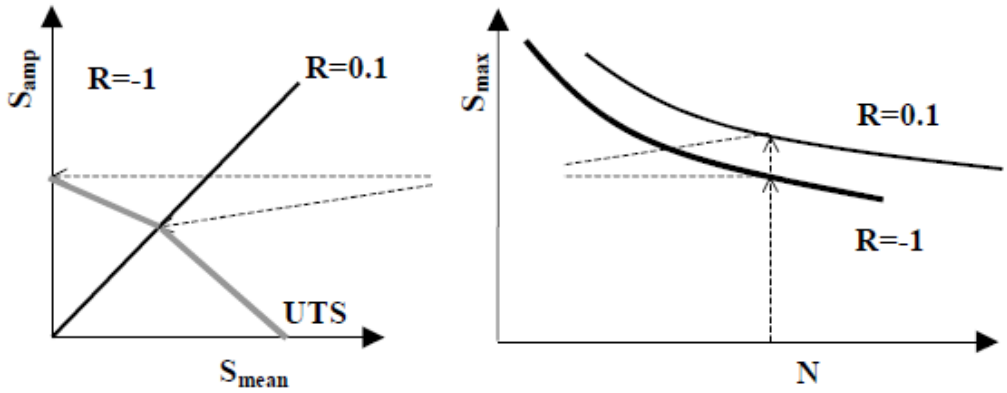


Fig. 2 (c): Curvas CLD

El proceso anterior lo hemos realizado para solicitaciones con carga media positiva, en el caso de carga media a compresión se haría lo mismo hacia la parte negativa del eje de carga media, con la diferencia de que en lugar de UTS tendríamos UCS (límite a compresión estático). Como en los *composites* el límite estático a compresión suele ser menor que el de tracción, los diagramas CLD no suelen ser simétricos.

La más conocida y empleada de estas curvas CLD debido a su simplicidad es el diagrama de Goodman. Para cualquier carga con una S_{media} y una $S_{amplitud}$, se obtiene una S_{equ} con R=-1.

En el diagrama de Goodman, como muestra la figura 2(d), tenemos dos áreas diferenciadas, una para deformación media a compresión y otra para la deformación media a tracción.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En el caso de estar a la derecha del eje de ordenadas, deformación media positiva, la fórmula a partir de la que se obtiene la deformación máxima equivalente es la (2.1).

$$S_{max,equ} = \frac{S_{amplitud} \cdot UTS}{UTS - S_{media}} \quad , \qquad (2.1)$$

Si por el contrario se está a la izquierda del eje de ordenadas será la ecuación (2.2), la que permita hallar la tensión máxima equivalente.

$$S_{max,equ} = \frac{S_{amplitud} \cdot UCS}{UCS - S_{media}} \quad , \qquad (2.2)$$

donde:

- UTS: Resistencia estática a tracción.
- UCS: Resistencia estática a compresión.

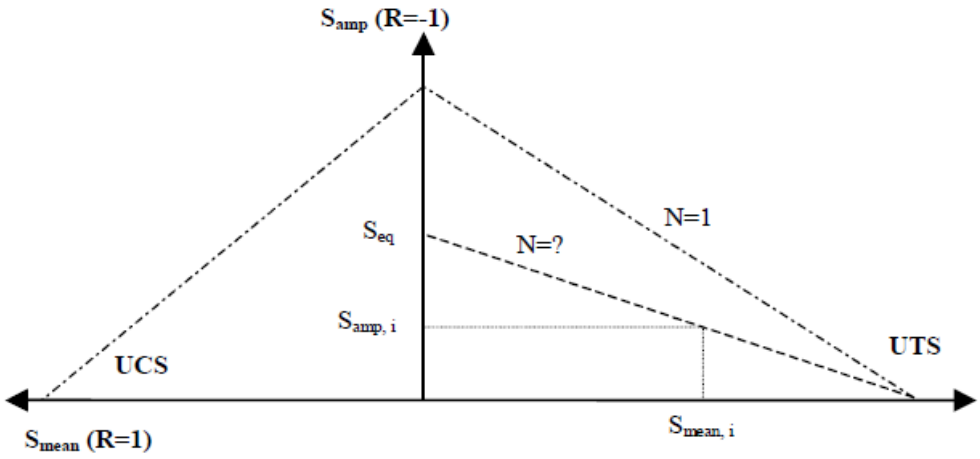


Fig. 2 (d): Diagrama de Goodman

En el diagrama de Goodman las líneas que forman un triángulo denotan puntos en función de amplitud de deformación y de deformación media con una vida estimada de un ciclo (N=1).

A partir de las ecuaciones (2.1) y (2.2), podemos obtener una deformación equivalente, cuya deducción gráfica (por semejanza de triángulos) se muestra en la Fig. 2 (d). Esta deformación equivalente es aquella que causa el mismo daño que la media y amplitud con la que hemos entrado, pero para una tensión media igual a 0, es decir R = -1. Por lo tanto, con la $S_{max,equ}$ entraré al diagrama *S-N* estimando la vida a fatiga (N).

Con esto lo que hemos ganado es que no necesitaremos un diagrama *S-N* para cada tipo de solicitación, sino que con el diagrama de Goodman y el diagrama *S-N* para R = -1, seré capaz de estimar la vida a fatiga del componente.

Sin embargo GL (ver [16]) en su última versión ha exigido emplear el diagrama de Goodman modificado. En éste, a diferencia del clásico, el máximo del diagrama se ha desplazado a la derecha, a

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

la media aritmética de UTS y UCS. Este desplazamiento a la derecha viene motivado por el hecho de que los materiales compuestos, a diferencia de los metales, soporten mucho mejor la tracción que la compresión.

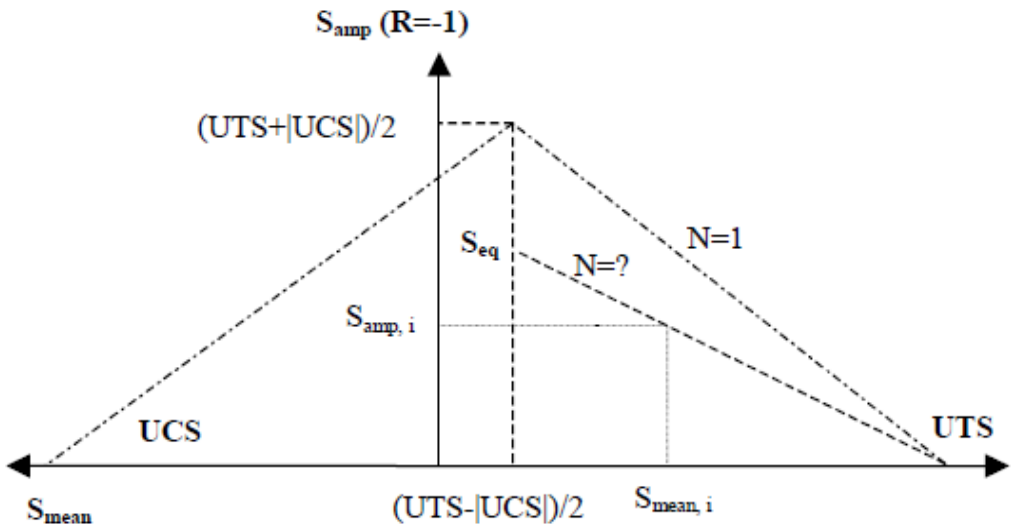


Fig. 2 (e): Diagrama de Goodman modificado

De lo anterior el número de ciclos permisibles a través de datos experimentales para R=-1 (sin incluir los factores de seguridad de GL) es:

$$N = \left[\frac{UTS + |UCS| - \left|2 \cdot S_{media} - UTS + |UCS|\right|}{2 \cdot S_{amp}}\right]^{\frac{1}{m}}$$

donde:

- N es el número de ciclos esperados al nivel de solicitación S_{media} y S_{amp} .
- UTS es el límite estático a tracción del material.
- UCS es el límite estático a compresión del material.
- S_{media} es la tensión o deformación media.
- S_{amp} es la amplitud de tensión o deformación.
- m es la pendiente de la curva *S-N*.

Esta formulación modificada de Goodman refleja la asimetría que se observa en los resultados de la experimentación, mientras que sigue manteniendo la simplicidad del diagrama de Goodman tradicional.

2.1.3.2 CARGA EQUIVALENTE

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Hay veces en las que es muy útil comparar los fallos de un diseño por distintos espectros de carga. Para ello se suele calcular una carga equivalente, que tenga amplitud constante con carga media igual a cero. Esta carga de amplitud constante sería aquella que causara el mismo daño al componente analizado que el espectro complejo de cargas.

Este método fue aplicado en palas de aerogeneradores por Brøndsted (ver [18]). El método de la carga equivalente se obtiene a partir del diagrama *S-N* y de la suma de Miner, explicada más adelante. Se asume que para cada ciclo de cargas *i* los parámetros m (pendiente del diagrama *S-N*) y $n_i \cdot S_i$, son constantes.

Al estar deducido a partir de la suma de Miner este método no tiene en cuenta la secuencia de carga. La deducción completa de la carga equivalente puede verse en [18].

$$S_{equ} = \left(\frac{\sum n_i \cdot {S_i}^m}{\sum n_i}\right)^{\frac{1}{m}} \cdot \left(\frac{1}{M}\right)^{\frac{1}{m}},$$

donde:

- S_{equ}: Es la tensión o deformación equivalente.
- S_i: Tensión o deformación máxima para los ciclos *i*.
- n_i: Número de ciclos con tensión o deformación *i*.
- M: Valor de la suma de Milner.
- m: Pendiente de la curva *S-N*.

2.1.4 RECUENTO DE CICLOS

Al presentar la fatiga hemos supuesto que las cargas de fatiga tenían una forma sinusoidal que se repetía continuamente. Como es de suponer, en la vida real las cargas que sufre un componente son por lo general de forma aleatoria y para nada regulares.

Lógicamente el estudio de cargas variables que no sigan ningún patrón es muy complejo, por lo que es imposible simular exactamente el espectro de cargas que sufre el componente. A partir de un espectro de cargas irregular debemos ser capaces de transformarlo en grupos, dentro de los cuales los ciclos de carga se repitan de forma regular. En esto es en lo que consiste el recuento de ciclos. Pasaremos a tener varios niveles de carga, cada uno de los cuales durará un número determinado de ciclos.

El objetivo es por tanto descomponer una historia de cargas, tensiones o deformaciones en una tabla de ciclos con S_{media} y $S_{amplitud}$.

Fundamentalmente hay tres procedimientos para realizar el recuento de ciclos:

1. *Level crossing:* Sólo se tienen en cuenta los ciclos cuyo pico supera un cierto nivel de carga tomado como referencia. Esto es interesante para materiales que como los aceros tienen límite a fatiga. En estos casos se toma como nivel de referencia el límite a fatiga del material.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

2. *Range-Mean:* Se divide el espectro de cargas en segmentos. Cada segmento está delimitado por un pico y su valle adyacente. A continuación, estos segmentos se caracterizan en términos de valor medio y de rango. Una vez hecho esto, se procede a contar los ciclos, tomando como medio ciclo cada uno de los segmentos.

3. *Rainflow counting:* Para reordenar la historia de cargas este método recorre el espectro de cargas y lo descompone en ciclos de carga completos y finalmente agrupa los que sean iguales. Este es el método más empleado.

Lógicamente, las historias de carga son muy complejas, por lo que estos procesos están implementados en programas de ordenador, como Twist o Enstaff, que ponen en práctica esos métodos.

Como *output* de este recuento de ciclos tenemos matrices de Markov. Estas matrices tienen en las columnas cargas de rango y en las filas cargas medias. Cada casilla contiene el número de ciclos para la carga media y de rango correspondiente a su fila y columna. Más adelante se da una explicación más detallada de lo que son las matrices de Markov y del uso que se hace de las mismas.

2.1.5 SUMA DE MINER

Las normativas de aerogeneradores recomiendan el uso de la suma de Miner para predecir la vida de los componentes ante una un espectro temporal de cargas.

La suma de Miner es una regla para calcular el daño acumulado por un componente debido a una historia de cargas. Éste daño no es un daño físico, es decir no es algo tangible.

La idea básica de este método es que cada ciclo con una determinada amplitud de carga y su correspondiente carga media causa un daño. Este daño se va sumando y acumulándose hasta que se alcanza el daño total estimado que puede soportar el componente. Estos daños se calculan como se ha explicado anteriormente con las curvas *S-N* y CLD. La expresión que describe este método se muestra a continuación:

$$D = \sum_i \frac{n_i}{N_i}$$

donde:

- D es el daño acumulado por el componente.
- n_i es el número de ciclos con carga de amplitud *i* y con carga media *i* a los que se somete el componente.
- N_i es el número de ciclos que sería capaz de soportar el componente antes de fallo con carga media y amplitud *i*.

Dado que la suma está normalizada se considera que el fallo se produce cuando el daño acumulado alcanza la unidad.

La suma de Miner tiene principalmente dos limitaciones:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- El daño no es un parámetro físico y no tiene que estar necesariamente relacionado con reducciones de rigidez o la resistencia. Además no hay consenso sobre la definición de fallo. En unos estudios se considera fallo la aparición de un grieta y en otros la separación completa de los componentes.

- El daño producido por cada ciclo es independiente del producido anteriormente. Es decir, no se tiene en cuenta la secuencia de cargas.

La precisión de la suma de Miner está también condicionada por la formulación escogida para las curvas *S-N* y CLD. Esto es debido a que el número de ciclos hasta el fallo para unas determinadas cargas media y de amplitud se determinan a través de las dos curvas mencionadas.

Por su simplicidad la regla de Miner es ampliamente usada en la práctica ingenieril. Sin embargo, debido a los condicionantes anteriormente expuestos puede llevar a imprecisiones. Estudios como [10] y [11], concluyen que la suma de Miner sobreestima la vida a fatiga por un factor de 2-3.

Para corregir algunos de estos errores se han introducido algunas modificaciones a la este método, apareciendo la “suma de Miner factorizada” o la “suma de Miner no lineal”. Información sobre estas alternativas se puede encontrar en diversa literatura como [19].

2.1.6 MODELO DE DEGRADACIÓN DE RESISTENCIA

Una alternativa a la suma de Miner para estimar la vida a fatiga de un componente consiste en estudiar la evolución de la resistencia del mismo. Se busca la degradación del material debida al daño por fatiga. Esta expresión tiene la ventaja de basarse en un daño físico, ser tangible, y de tener en cuenta la secuencia de cargas.

Hay muy diversos modelos según diferentes autores, pero todos ellos se caracterizan por estimar la resistencia residual que queda tras un número determinado de ciclos. La degradación de la resistencia ocurre en tres fases:

- Inicial: Fase no lineal de corta duración.

- Intermedia: Lineal, es la que más dura.

- Final: No lineal, durante esta fase se produce el fallo de manera repentina.

No se va a profundizar en esta metodología, porque si bien es más completa no es la que se aplica en el campo que nos concierne. Es cierto que los centros de ensayos proporcionan información al respecto al realizar un ensayo a fatiga, pero el presente proyecto tiene que ver exclusivamente con el preproceso del ensayo y ahora mismo no se tienen en cuenta ese tipo de datos.

2.2 FATIGA EN PALAS DE AEROGENERADOR

En relación con la fatiga de los aerogeneradores, las palas actuales tienen dos características principales:

- Están sometidas a un gran número de ciclos de carga a lo largo de su vida útil.

- Están formadas por complejas estructuras de materiales compuestos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Éstas características condicionan el estudio a fatiga en este campo. Primero, porque deberán soportar un gran número de ciclos y, segundo, porque deberemos centrarnos en la fatiga relativa a los materiales compuestos.

En la siguiente figura se muestran las principales razones por las cuales la fatiga es tan crítica en las palas de los aerogeneradores.

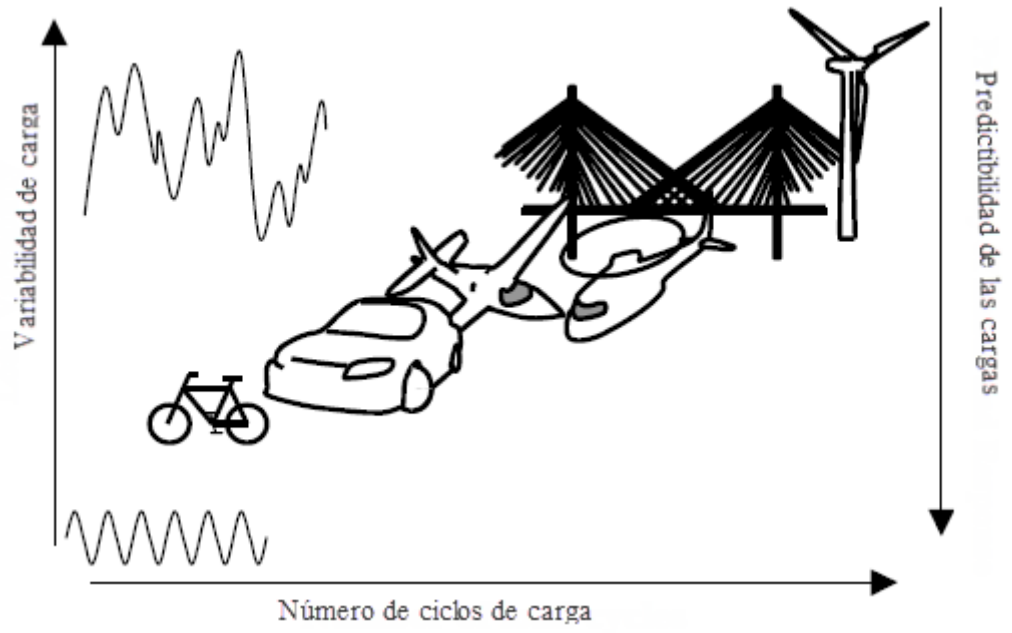


Fig. 2 (f): Régimen de carga de aerogeneradores

La figura denota tres razones que justifican la criticidad de la fatiga en las palas de los aerogeneradores:

- El número de ciclos de carga durante la vida útil es mucho más elevado que en la industria aeronáutica o en la automovilística.
- La variabilidad de las cargas a que se ven sometidas es mucho mayor que en cualquier otro caso.
- La predictibilidad de las cargas es muy baja.

La vida requerida para un aerogenerador es de veinte años, los cuales vienen a suponer aproximadamente 10^8 ciclos de carga. Estos ciclos se ven reflejados en las matrices de Markov.

Tanto la variabilidad de las cargas como su baja predictibilidad se deben a la naturaleza estocástica del viento, que lo hace variable e imprevisible. No olvidemos que, dependiendo de la velocidad del mismo, el aerogenerador arrancará o se detendrá. Además, de que las cargas sufridas por el mismo dependen de la velocidad del viento.

Las cargas en los aerogeneradores suelen descomponerse en cargas en la dirección de *flap* y cargas en la dirección de *edge* (Fig. 2 (g)). A pesar de los continuos esfuerzos por integrar la optimización estructural y aerodinámica, las geometrías actuales de las palas se pueden dividir en componentes de

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

naturaleza estructural y partes optimizadas geométricamente para cumplir requisitos aerodinámicos. Las cargas de *flap* son soportadas principalmente por la viga de la pala, mientras que las de *edge* lo son por la viga y por refuerzos que pudieran colocarse en bordes de ataque y de salida. En la figura 2 (g) también se muestran las cargas soportadas por cada parte de la sección.

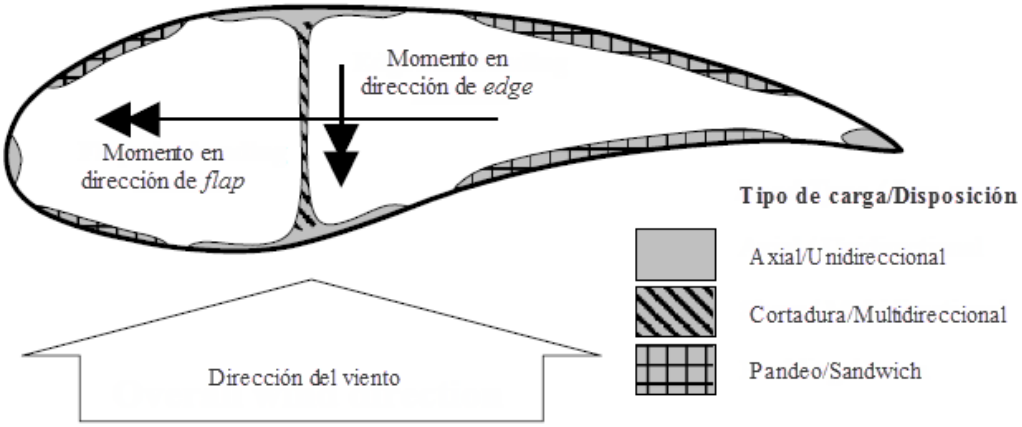


Fig. 2 (g): Sección típica de una pala.

Las cargas de *flap* son causadas por el viento que incide perpendicularmente al área barrida por las palas. Dado que el viento varía continuamente, su variabilidad es muy grande. Las cargas de *edge* son principalmente causadas por el peso propio de la pala y por el torsor que hace girar a la pala. Éstas últimas varían dos veces por vuelta, con lo que su variabilidad es menor.

Las cargas de fatiga en las palas de los aerogeneradores tienen un cierto grado de variabilidad estadística. Entre la variabilidad de las propiedades de los materiales y otras incertidumbres, la indeterminación de las cargas de fatiga es una fuente de dispersión. Con ello, el modelo que se elija para obtener las cargas puede producir importantes variaciones en la fiabilidad a fatiga.

2.2.1 OBTENCIÓN DE LAS CARGAS DE FATIGA

El espectro completo de cargas de fatiga se debe construir a partir de distintos espectros referentes a los diferentes modos de operación del aerogenerador.

Por un lado, se debe hacer durante una hora una simulación aerodinámica para cada velocidad de viento media a la que se va a ver sometido. Estas simulaciones se llevan a cabo mediante softwares apropiados. De estas simulaciones se obtienen las cargas cíclicas que va a sufrir el aerogenerador durante una hora para cada velocidad de viento. Por lo tanto, multiplicando estos datos por el número de horas que se estima que va a operar la pala a cada velocidad de viento, obtenemos las cargas que sufrirá la pala para cada velocidad de viento durante su vida. La normativa europea IEC-61400, especifica que las estimaciones del viento deben de hacerse de acuerdo con la distribución de Rayleigh, con la velocidad media anual de viento especificada en la citada normativa para la región de interés (ver [15]).

Por otro lado, también se debe obtener el espectro de cargas debido a que la máquina esté parada o en mantenimiento y también a los arranques, paradas ordinarias y paradas de emergencia.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A todos los espectros de carga se les debe aplicar el método de recuento de ciclos *Rainflow counting* explicado anteriormente. Para calcular la vida de la pala se deberá tener en cuenta tanto los ciclos por los arranques y paradas como los obtenidos con las distintas velocidades medias de operación.

En la figura 2 (h) se muestran dos espectros típicos de cargas a fatiga en una pala. En el lado izquierdo aparece un espectro típico de *flap* y a la derecha uno típico de *edge*.

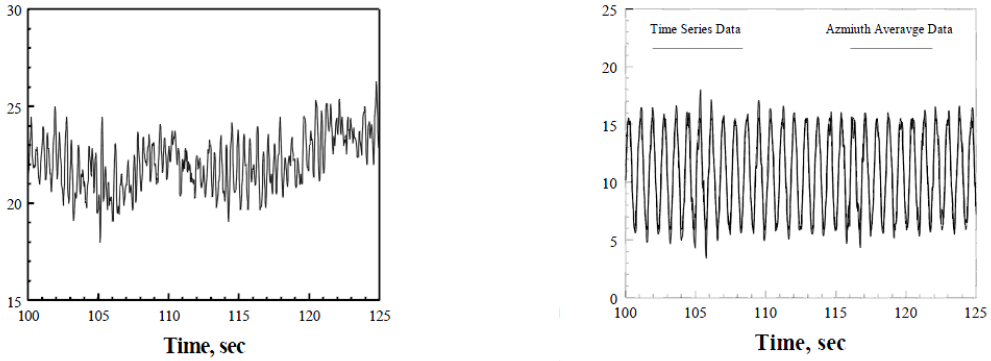


Fig. 2 (h): Espectros temporales de *flap* y *edge*.

Como hemos dicho, una vez que disponemos de los distintos espectros de carga que tenemos que tener en cuenta, les aplicamos mediante un software el método de *Rainflow counting*. Como ya explicamos, este método descompone el espectro temporal agrupándolo en grupos de ciclos con la misma carga media y carga de rango.

Esos grupos de ciclos con las mismas cargas se almacenan en lo que se conoce como matrices de Markov. En la figura 2 (i) se puede ver una matriz de Markov dibujada en tres dimensiones. Estas matrices almacenan todo el espectro de cargas al que se va a someter una pala a lo largo de su vida.

Dado que las dimensiones de la pala son muy grandes (pueden llegar a unos 60 metros), las cargas a lo largo de la misma varían y no es realista tener una sola matriz de Markov para toda la pala. Por lo tanto, se discretiza la pala y se obtienen las matrices de Markov para una serie de secciones discretas de la pala.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

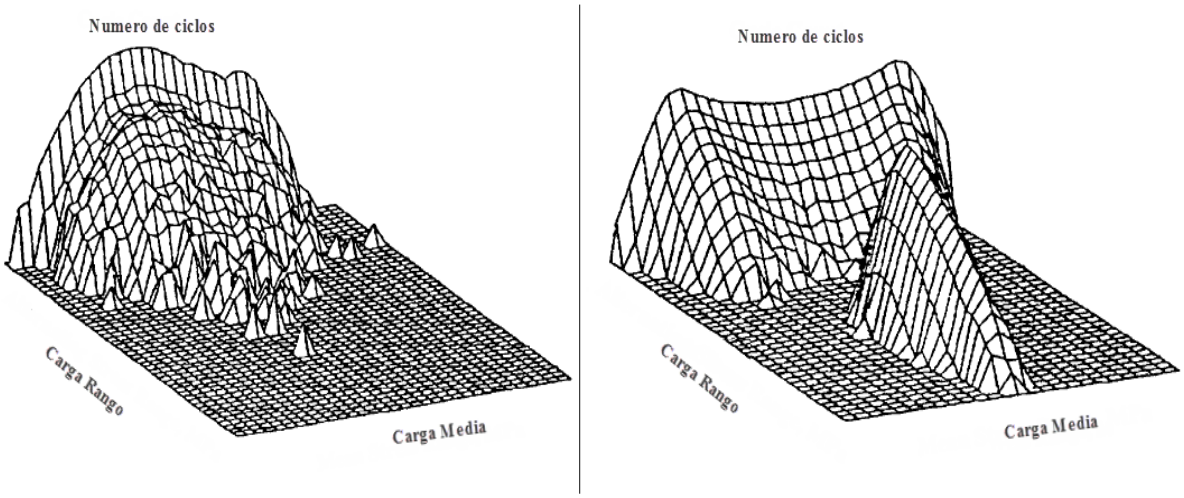


Fig. 2 (i): Matrices de Markov *flap* (izquierda) y *edge* (derecha)

Una vez disponemos de las matrices de Markov podemos calcular, mediante la suma de Miner el daño sufrido por la pala durante toda su vida útil. Lógicamente para poder llevar a cabo la suma de Miner debemos haber calculado el daño que causa cada una de las casillas de la matriz de Markov. Para ello debemos aplicar según GL (ver [16]), el diagrama *S-N* y el diagrama de Goodman modificado. GL nos da directamente la fórmula para calcular los ciclos, N_i (ver Fig.: 2(j)) que puede resistir la pala para una determinado nivel de carga *i* (media y de rango). Como el número de ciclos a cada nivel de carga, es el que aparece en la casilla de la matriz de Markov correspondiente, la suma de Miner es ya directa. Recordar:

$$D = \sum_i \frac{n_i}{N_i}$$

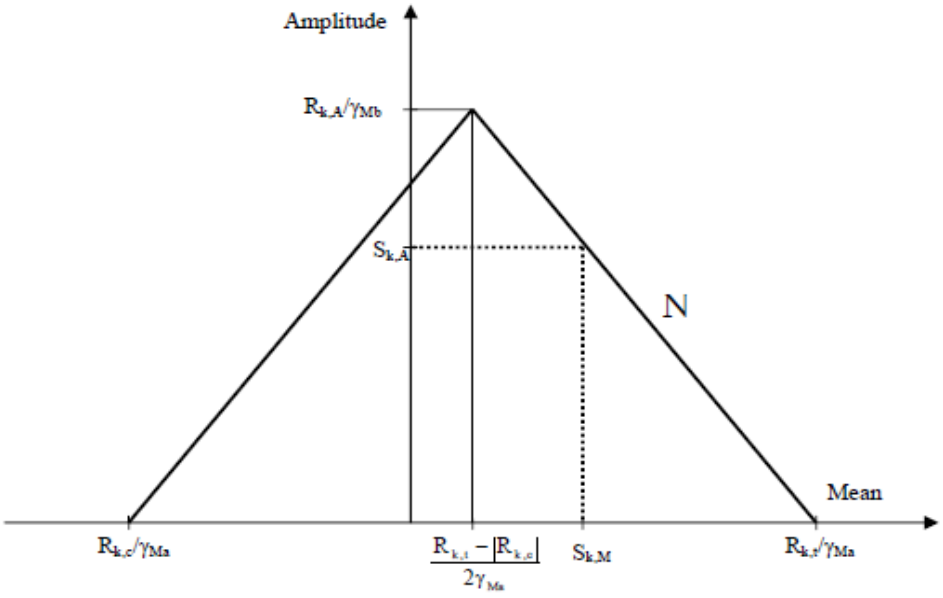


Fig. 2 (j): Diagrama de Goodman modificado proporcionado por GL

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

El diagrama de Goodman modificado anterior lleva incorporados los coeficientes de seguridad γ_{Ma} y γ_{Mb} que marca GL (ver [16]). A partir del mismo y del diagrama *S-N*, se obtiene la siguiente fórmula, también proporcionada por GL:

$$N = \left[\frac{R_{k,t}+|R_{k,c}|-\left|2\cdot\gamma_{Ma}\cdot S_{k,m}-R_{k,t}+|R_{k,c}|\right|}{2\cdot\left(\frac{\gamma_{Mb}}{C_{1b}}\right)\cdot S_{k,a}}\right]^m,$$

donde:

- N: Es el número de ciclos tolerables antes del fallo.
- $S_{k,m}$: Valor medio de las tensiones o deformaciones.
- $S_{k,a}$: Amplitud de tensiones o deformaciones.
- $R_{k,t}$: Resistencia estática a tracción en forma de tensiones o deformaciones.
- $R_{k,c}$: Resistencia estática a compresión en forma de tensiones o deformaciones.
- m: Pendiente de la curva *S-N*.
- γ_{Ma}: Factor de seguridad estático para el material.
- γ_{Mb}: Factor de seguridad de cargas dinámicas para el material.

2.2.2 ENSAYOS A FATIGA

En la India, Alemania y Dinamarca, un aerogenerador debe estar certificado para que se le conceda el permiso de construcción. En la mayor parte de Europa, EEUU y Canada, los bancos y grupos inversores exigen que los aerogeneradores estén certificados.

Por lo tanto, los aerogeneradores y, por ende, las palas, deben pasar una serie de ensayos. Concretamente, las palas deben pasar un ensayo *full-scale* en base a unas cargas de diseño en un centro de ensayos reconocido y bajo la supervisión de una entidad certificadora.

Para conseguir la certificación, las palas deberán soportar los ensayos estáticos y de fatiga, sin que se registre ningún daño que comprometa su funcionalidad. Las principales entidades certificadoras son DNV, IEC, DS (Danish Standard Foundation) y GL.

Los ensayos de palas no siempre tienen por qué tener fines certificativos. Si bien debido a su elevado coste la empresa trata de evitar realizar ese tipo de ensayos, muchas veces en el periodo de desarrollo de nuevos modelos la empresa necesita testar como funcionan sus nuevos desarrollos, razón por la cual deberá hacer ensayos.

El ensayo de fatiga *full-scale* es obligatorio sólo según IEC (ver [15]) y DS (ver [17]).

Para pasar el ensayo de fatiga se requiere que la pala sea ensayada con unas cargas suficientemente superiores a las de diseño. Además, de acuerdo con IEC (ver [15]) y con DS (ver [20]), las cargas de

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

diseño deben ser multiplicadas por un coeficiente de seguridad de 1.3285. Una vez multiplicada por ese coeficiente la carga resultante recibe el nombre de carga de ensayo:

$$Carga\ de\ ensayo = 1.3285 \cdot Carga\ de\ diseño.$$

El coeficiente 1.3285 incluye:

- $\gamma_{nf} = 1.15$: Factor de seguridad por las consecuencias de fallo.
- $\gamma_{sf} = 1.1$: Variabilidad de pala a pala (no son todas iguales).
- $\gamma_{ef} = 1.05$: Errores en la formulación de fatiga.

El factor de seguridad por variabilidad de una pala a otra puede parecer extremadamente conservador¸ pero debe tenerse en cuenta que en la actualidad las palas se manufacturan de manera muy artesanal y poco automatizada. Muchas de las telas de material compuesto son colocadas a mano por personas. Esto conlleva inevitables errores humanos, que dan lugar a diferencias considerables entre dos especímenes que según diseño debieran de ser iguales.

Tal como se ha explicado en el apartado anterior, se dispone de las cargas de diseño almacenadas en las llamadas matrices de Markov. Además disponemos de una matriz de Markov para una serie de secciones discretas de la pala. Sin embargo no es factible realizar un ensayo representando todos los ciclos que aparecen en la matriz, tanto por el número de ciclos como por la dificultad de representar tal variabilidad de ciclos.

Lo que se debe hacer es determinar uno o varios niveles de carga, que durante 10^6, $2 \cdot 10^6$ o $5 \cdot 10^6$ ciclos, causen el mismo daño que el de diseño representado por las matrices de Markov. Generalmente con uno o dos niveles de carga es suficiente, dado que a más niveles de carga aumentaremos la complejidad del ensayo. Si bien es verdad que a más niveles de carga mejor representaremos el espectro real de cargas. Recordar que el daño representado por las matrices de Markov es el estimado para la vida del aerogenerador. Por lo tanto, si conseguimos diseñar un ensayo en el que se le cause al aerogenerador tanto daño como el de las matrices de Markov, estaremos “asegurándonos” que nuestra pala va a ser capaz de no fallar por fatiga antes de los veinte años para los que se ha diseñado.

A la hora de elegir el nivel de carga y el número de ciclos al que se va a ensayar no hay pautas claras y dependerá de la habilidad del ingeniero el hacer un ensayo que se ajuste al máximo posible a la realidad.

Hay quien sólo se fija en el daño causado sin importar que el nivel de cargas no se parezca en nada al espectro real. La normativa no dice nada en contra de ello. Sin embargo, si se quiere evolucionar en la investigación de la fatiga, se debe tender a buscar ensayos que, además de causar el mismo daño, consideren niveles de carga parecidos al espectro real.

Es cierto que en el espectro real hay infinidad de ciclos, lo que a priori dificulta representar un ensayo que se asemeje a este espectro. Sin embargo, la mayor parte del daño es causada solo por unos pocos ciclos, mientras que la mayor parte de los ciclos que aparecen en la matriz de Markov causan un daño despreciable.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

En la industria aeronáutica, donde el análisis de la fatiga es una parte fundamental del diseño, se trata de conseguir la mejor aproximación posible a las cargas del espectro real. Para ello, se filtran los ciclos que no causan daños significativos, quedándose solo con los más representativos del daño causado. Esto es equivalente a quedarse solo con las celdas de la matriz de Markov más representativas, lo que hace que el espectro real de cargas se vea reducido considerablemente.

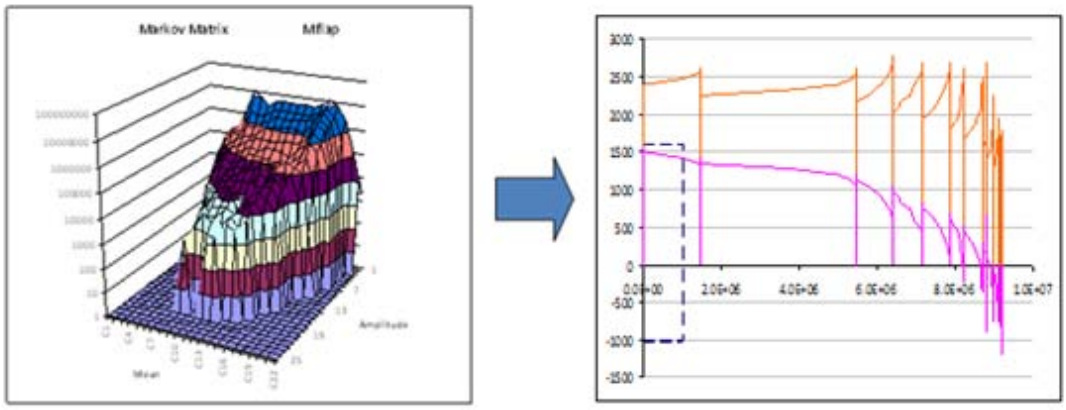


Fig. 2 (k) : Espectro real de cargas

En la figura anterior se ve como dibujamos la matriz de Markov como cargas máximas y medias frente a número de ciclos. Estos serán filtrados para quedarse, como se ha explicado, solo con los ciclos que causan un daño significativo. El filtrado de ciclos da lugar a la figura 2 (l).

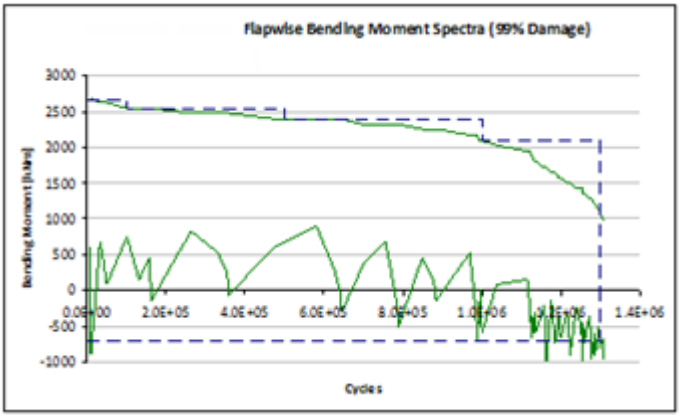


Fig. 2 (l): Truncado de ciclos y diseño por niveles de carga.

Tras hacer esto la duración del ensayo queda establecida. Dicha duración será similar al número de ciclos que quedan tras haber eliminado las casillas de la matriz de Markov que causan un daño despreciable.

Establecido el número de ciclos para el ensayo, el paso siguiente es decidir los niveles de carga. A más niveles de carga más ajuste pero mayor complejidad, lo que nos obliga a llegar a un compromiso. Teniendo en cuenta que los *composites* sufren más a fatiga con cargas medias negativas, como carga mínima cogeremos aquella que envuelva el espectro real, como se observa en la figura 2 (l). Y como cargas máximas iremos cogiendo niveles de carga por escalones de manera que aproximemos espectro real filtrado de cargas máximas (ver Fig. 2 (l)).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por último, ya solo nos quedaría asegurarnos de que el daño causado según la regla de Miner, es el mismo o superior al que causaba la matriz de Markov original.

Lo anterior se aplica a cada sección en la que tenemos definido el espectro de cargas. Esos niveles de carga se traducen en una carga de rango y una carga media para cada nivel de carga y cada radio. Dado que, alcanzando en esas secciones esa carga media (momento medio) y esa carga de rango (momento de rango), estaremos cumpliendo con los requisitos de diseño, esta distribución de cargas de rango y media, será nuestra distribución objetivo.

Como en el caso que nos concierne las cargas son momentos, lo que nosotros tenemos es una distribución de momentos medios objetivo y de momentos de carga objetivo, definida para una serie de secciones. Ésta será la distribución que deberemos aproximar al máximo en el ensayo a fatiga.

Como se ha visto con anterioridad, los aerogeneradores se diseñan para que tengan una vida útil de veinte años. Es obvio que las empresas no se pueden permitir el estar veinte años ensayando un modelo antes de ponerlo en el mercado. Primero, porque a los veinte años ese diseño estaría ya obsoleto y, segundo, por razones de costes.

Para poder acortar los tiempos de ensayo hay dos variantes, que no hacen sino acelerar el esnsayo por dos vías distintas:

1. Aumentar la frecuencia.

2. Aumentar las cargas de ensayo.

De hecho lo aconsejable no es usar una vía u otra sino combinar ambas técnicas. Dado que las cargas de fatiga, si bien no son tan elevadas como las estáticas, tienen un valor considerable, para aumentarlas y ejercerlas cíclicamente necesitaríamos gran cantidad de energía y una maquinaria con altas prestaciones. Es por ello que comúnmente, en los centros de ensayos, se excita la pala según su primera frecuencia natural de manera que la pala entra en resonancia. Al entrar en resonancia, las vibraciones son mucho más elevadas de lo normal, lo que provoca la aparición de unas fuerzas de inercia que son muy superiores a las ejercidas por el excitador. Este fenómeno de resonancia nos facilita conseguir aumentar las cargas considerablemente sin necesidad de emplear unas máquinas con unos brazos enormes.

Si se excita una estructura, en este caso una pala, en su primera frecuencia natural de vibración¸ ésta vibra deformándose según su modo de vibración correspondiente.

Para excitar la pala se utilizan unos excitadores o actuadores que pueden ser de dos tipos:

- Actuadores hidráulicos: Consisten en una masa oscilatoria que se desplaza a través de una carrera a la frecuencia oscilatoria.

- Masa rotatoria: Se coloca una masa rotatoria en un punto de la pala, haciéndola girar a la frecuencia seleccionada. Este método es el empleado por la escuela danesa.

Emplear como actuador una masa rotatoria presenta el problema de inducir cargas no deseadas. Esto es debido a que, además de generar una carga en la dirección de *flap* o de *edge*, según el ensayo, generamos una fuerza en la dirección axial. En la figura 2 (m) tenemos un dibujo esquemático de un ensayo a fatiga con un actuador de masa rotatoria y cuatro masas muertas. Notar que, por razones de

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

seguridad la masa giratoria está colocada dentro de una jaula que se ubica en el punto deseado de la pala.

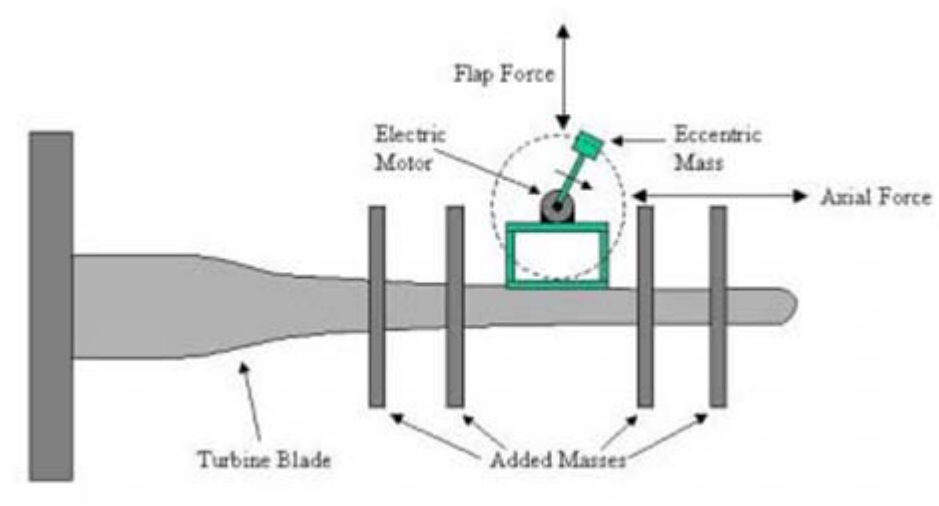


Fig. 2 (m): Ensayo a fatiga con excitador de masa rotatoria.

Por otra parte, los actuadores hidráulicos son masas concentradas accionadas hidráulicamente que oscilan desplazándose sobre unas guías de un conjunto colocado en la pala, en la sección seleccionada. En la figura 2 (m) tenemos una pala con un tipo de actuador hidráulico, mientras que en la figura 2 (n) tenemos la imagen de otro tipo de actuador hidráulico con el soporte que se colocaría sobre la pala.

Los actuadores ejercen una fuerza sobre la pala, pero esa no es su misión principal, puesto que las fuerzas de inercia generadas en la pala al hacerla vibrar con su frecuencia natural son mucho mayores. Por tanto la función principal de estos actuadores es excitar la pala según su primera frecuencia natural de vibración.

Es importante señalar que una distribución de desplazamientos de las diferentes secciones de la pala, implica directamente una distribución de momentos, porque como se explica más adelante:

$$M(x) = E(x) \cdot I(x) \cdot \frac{\partial^2 u(x)}{\partial x^2}$$

siendo:

- M [Nm] es el momento flector.
- E [N/m^2] el módulo de Young.
- I [m^4] es el momento de inercia.
- u [m] es el desplazamiento vertical.
- x [m] es la variable longitudinal de la viga.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

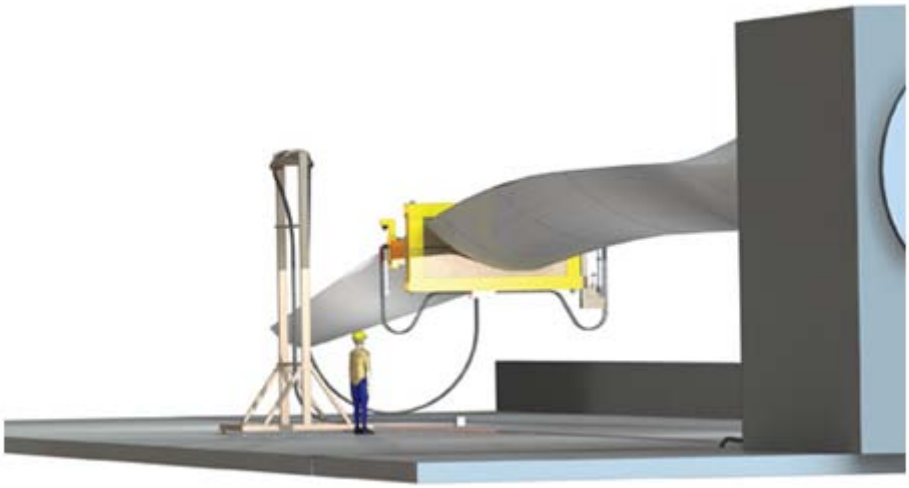


Fig. 2 (n): Actuador hidráulico para ensayo a fatiga.

Fig. 2 (o): Actuador hidráulico con su soporte correspondiente.

Si excitásemos la pala según su primer modo de vibración y tuviésemos la suerte de que con este modo la pala se deformase con una distribución de desplazamientos que implicase la distribución de momentos objetivo, habríamos terminado. Lógicamente eso no suele darse.

Debemos recordar que nuestros momentos objetivo son, por un lado, una distribución de momentos medios y, por otro, una distribución de momentos de rango. Cuadrar las dos al mismo tiempo suele ser un tema bastante complejo. Por todo esto, para conseguir alcanzar los momentos objetivo además de los actuadores se añaden una serie de masas muertas a lo largo de la pala (ver Fig. 2(q)). Estas masas muertas nos deben ayudar a alcanzar tanto la distribución de momentos medios, así como la de momentos de rango.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Fig. 2 (q): Masa muerta colocada en una pala.

En función de la masa que añadamos y de la sección de la pala en la que la añadamos, modificaremos el modo de vibración de la pala. Dado que en el ensayo a fatiga hacemos vibrar a la pala según su primer modo de vibración y puesto que cada deformada implica una distribución de momentos (se verá más adelante), la distribución de masas muertas nos permitirá modificar la distribución de momentos rango alcanzada.

A su vez la pala tiene una deformada debida a su peso propio, que conlleva una distribución de momentos medios. Al modificar las masas muertas y sus posiciones estaremos variando esa deformada y, por ende, la distribución de momentos medios.

En la figura 2 (q), se ve una pala en un ensayo a fatiga con un excitador. El excitador es la jaula gris que se observa encima de la pala. En este caso es un excitador de masa rotatoria. Además la pala tiene colocadas dos masas muertas cerca de la punta y otra más antes del actuador.

Fig. 2 (q): Ensayo a fatiga con un actuador y tres masas muertas.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Hasta ahora lo frecuente era emplear un solo excitador para los ensayos de las palas. Sin embargo, ante la evolución hacia palas de dimensiones mucho mayores, el empleo de un solo excitador hace imposible ajustar la distribución de momentos objetivo a lo largo de toda la pala. Es por esta razón que los centros de ensayo han empezado a trabajar con más de un excitador, para ser capaces de, en un mismo ensayo, evaluar toda la pala.

Sin embargo, empieza a haber centros de ensayo que no se conforman con solo dos excitadores, sino que con técnicas novedosas llegan a incorporar hasta cinco excitadores en función de las dimensiones de la pala a ensayar.

La tendencia a desarrollar palas más grandes no solo implica el que hagan falta más excitadores, sino que al ser palas más pesadas y grandes su primera frecuencia natural de vibración baja ostensiblemente. Esto trae consigo la problemática de que si los ensayos a fatiga ya tenían una duración considerable, se alargan todavía más, lo que aporta una complicación tanto de costes como de viabilidad.

Para certificar una pala se deben hacer dos ensayos a fatiga uno en la dirección de *flap* y otro en la de *edge*. Ante esta problemática, lo que los centros de ensayos están proponiendo es realizar simultáneamente los dos ensayos, es decir cargar la pala biaxialmente. Esto reduciría costes y tiempo haciendo el ensayo mucho más viable.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

3 DEDUCCIÓN DE LA ECUACIÓN BIARMÓNICA DE ONDAS

Para deducir la ecuación que modeliza el problema que nos concierne realizamos una serie de suposiciones de acuerdo con la teoría de Euler-Bernoulli (ver [12]):

- Comportamiento elástico: el material de la viga es elástico lineal, con módulo de Young E y coeficiente de Poisson despreciable, es decir, no alcanza régimen plástico.
- Flecha vertical: en cada punto de la viga el desplazamiento vertical solo depende de x (coordenada longitudinal).
- Las secciones planas para la viga sin deformar siguen siendo planas para la viga deformada.
- Sólo se tiene en cuenta la deformación por momento flector, despreciándose la deformación por cortante (las palas son muy esbeltas) y no se consideran esfuerzos normales a la sección estudiada.
- Las deformaciones son lo suficientemente pequeñas para que la acción de las fuerzas externas no se vea modificada, en primera aproximación, por la deformación.

Tomando como modelo para describir el comportamiento de la pala la viga de Euler-Bernoulli, pasamos a deducir la ecuación que modeliza su movimiento.

Para obtener la ecuación de la elástica supondremos una viga sometida a flexión y cuya deflexión es solo debida al momento flector, como es el caso que nos ocupa.

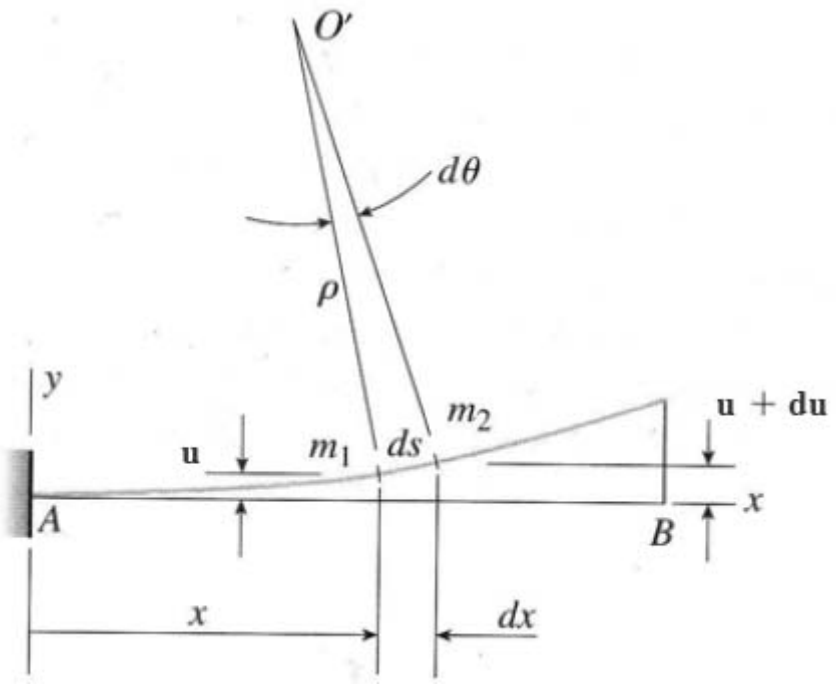


Fig. 3 (a) : Viga sometida a flexión.

La deflexión en el punto m_1 es la distancia a éste desde al eje de abcisas, u. Si tomamos un punto m_2 infinitesimalmente próximo al anterior (su abcisa será x+dx), su deflexión habrá variado otra cantidad

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

infinitesimal, du, que se corresponde con el incremento de la deflexión conforme avanzamos a lo largo de la curva de m_1 a m_2. La deflexión de este segundo punto será u+du.

Al flexionarse la viga cada uno de sus puntos realiza dos movimientos:

- Se desplaza una cierta cantidad u.
- Gira un cierto ángulo.

Llamaremos ángulo de rotación, θ, del eje de la viga al ángulo formado por la horizontal y la tangente a la elástica en ese punto. Así, el punto m_1 tendrá un ángulo θ, mientras que el punto m_2 habrá girado un ángulo θ+dθ.

Por otro lado, si como en la figura, trazamos las normales a las tangentes de los dos puntos, dichas normales formarán entre si un ángulo dθ, cortándose en el punto O'. Dicho punto recibe el nombre de centro de curvatura, mientras que la distancia de O' a la curva se le denomina radio de curvatura, ρ.

Al ser la longitud del arco igual al radio por el ángulo:

$$ds = \rho \cdot d\theta,$$

donde ds es la distancia sobre la curva que separa los dos puntos. Entonces κ que es la inversa del radio de curvatura, se expresa mediante la ecuación:

$$\kappa = \frac{1}{\rho} = \frac{d\theta}{ds}. \quad (3.1)$$

Puesto que, según las hipótesis de Euler-Bernoulli, se suponen pequeñas deformaciones:

$$\theta \approx \tan\theta \qquad ds \approx dx .$$

Además, por la definición de tangente:

$$\theta \approx \tan\theta = \frac{du}{dx} .$$

Reescribiendo la ecuación (3.1), obtenemos:

$$\kappa = \frac{1}{\rho} = \frac{d\theta}{dx} = \frac{d}{dx}\left(\frac{du}{dx}\right) = \frac{d^2u}{dx^2} \Longrightarrow \frac{1}{\rho} = \frac{d^2u}{dx^2} . \quad (3.2)$$

Como según la ley de flexión de Navier (ver [12]):

$$\frac{1}{\rho} = \frac{M}{E(x) \cdot I(x)} . \quad (3.3)$$

Igualando las ecuaciones (3.2) y la (3.3) obtenemos:

$$M = E(x) \cdot I(x) \cdot \frac{d^2u(x)}{dx^2} . \quad (3.4)$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Visto esto, lo siguiente es analizar una sección cualquiera de la viga (en nuestro caso de la pala). Para ello realizaremos un análisis dinámico de una sección de viga cualquiera de longitud dx. En concreto, partiremos de un análisis estático para luego añadirle la parte dinámica.

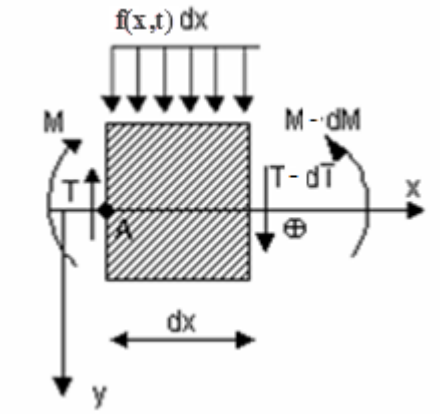


Fig. 3 (b) : Análisis estático sección de viga.

Al estar la viga en equilibrio, si aplicamos un simple equilibrio estático de fuerzas verticales (F_v) y de momentos respecto al punto A (M_A), sus respectivas sumas deben ser iguales a 0:

$$\sum F_v = 0 \rightarrow -T + T - dT + f \cdot dx = 0 \rightarrow \frac{dT}{dx} = f \ ,$$

$$\sum M_A = 0 \rightarrow (-M - dM) + f \cdot dx \cdot \frac{dx}{2} + M + (T - dT) \cdot dx = 0 \rightarrow \frac{dM}{dx} = T \ ,$$

donde:

- M [Nm] es el momento.
- T [N] es el cortante.
- f [N/m] es la fuerza por unidad de longitud.
- F_v [N] es la fuerza vertical.
- M_A [Nm] es el momento respecto al punto A.

De este equilibrio se deduce que:

$$f = \frac{d^2M}{dx^2} \quad . \quad (3.5)$$

A partir de las ecuaciones (3.4) y (3.5) obtenemos la fuerza por unidad de longitud debido al momento flector existente en la viga:

$$f = \frac{\partial^2}{\partial x^2}\left(E(x) \cdot I(x) \cdot \frac{\partial^2 u(x)}{\partial x^2}\right) \quad (3.6)$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Recordando la segunda ley de Newton, ésta nos dice que las fuerzas inerciales son igual al producto de la masa por la aceleración. De todos modos, como en el equilibrio estático hemos obtenido la fuerza por unidad de longitud, el cálculo de la fuerza inercial también lo realizaremos por unidad de longitud. Para ello, simplemente, multiplicaremos la masa por metro de viga por la aceleración.

Por lo tanto, la fuerza por unidad de longitud actuando sobre el elemento será igual a la suma de las fuerzas debidas al momento flector más las fuerzas inerciales. Con ello, obtenemos la ecuación diferencial que describe el problema que abordamos y que se muestra a continuación:

$$\frac{\partial^2}{\partial x^2}\left(E(x)\cdot I(x)\cdot\frac{\partial^2 u(x,t)}{\partial x^2}\right)+m(x)\cdot\frac{\partial^2 u(x,t)}{\partial t^2}=f(x,t), \quad (3.7)$$

donde:

- E [Nm^{-2}] : Módulo de Young.
- I [m^4] : Momento de inercia respecto a eje z.
- u [m] : Desplazamientos de cada punto en la dirección de acción de la carga.
- t [s] : Tiempo.
- x [m] : Dimensión longitudinal de la pala.
- m [kg/m] : Masa por unidad de longitud.
- f [N/m] : Fuerza por unidad de longitud.

Dado que en el diseño de palas se suele trabajar directamente con rigideces en lugar de hacerlo con el módulo de Young (E) y el momento de Inercia (I), denotaremos por D la rigidez de la pala que, en cada punto de la misma, queda definida como sigue:

$$D(x)=E(x)\cdot I(x)$$

Por lo tanto, la ecuación dinámica de la pala podría reescribirse de la siguiente forma:

$$\frac{\partial^2}{\partial x^2}\left(D(x)\cdot\frac{\partial^2 u(x,t)}{\partial x^2}\right)+m(x)\cdot\frac{\partial^2 u(x,t)}{\partial t^2}=f(x,t) \quad (3.8)$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

4 DESARROLLO DEL TRABAJO

4.1 INTRODUCCIÓN

El software desarrollado se divide en cuatro bloques claramente diferenciados. En cada uno de esos bloques se ejecutan una serie de instrucciones, que conjuntamente, permiten lograr el objetivo del programa.

En los siguientes cuatro apartados se describen las acciones realizadas y los resultados obtenidos en cada uno de los bloques, sin entrar en el detalle de los métodos matemáticos aplicados. Dichos métodos serán analizados más adelante en sus apartados correspondientes.

4.2 PROPIEDADES DE LA PALA

El primer bloque del programa, "Propiedades de la pala", es el más somero y sencillo de toda la aplicación. En él simplemente se cargan las propiedades de la pala necesarias para simular el ensayo. Por tanto es el punto donde se introducen los inputs del programa relativos a las propiedades de la pala a ensayar.

Para tener físicamente caracterizada la pala, necesitamos:

1. Radios en los que las propiedades son conocidas.
2. Masa lineal en cada uno de los radios.
3. Rigidez en *flap* en cada uno de los radios.
4. Rigidez en *edge* en cada uno de los radios.
5. Cuerda de la pala en cada uno de los radios.
6. Espesor de la pala en cada uno de los radios.

Todos los datos deberán ser introducidos en las unidades del sistema internacional.

En la siguiente figura se muestra esquemáticamente cuales son las direcciones de *edge* y de *flap*, en una pala.

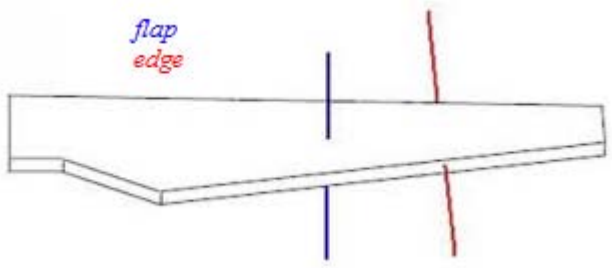


Fig. 4 (a): Direcciones de *flap* y *edge*.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Dado que una de las opciones que ofrece el programa es simular el ensayo a fatiga en la dirección de *flap* o bien simularlo en la dirección de *edge*, el usuario debe introducir tanto las rigideces en la dirección de *edge* como en la de *flap*.

Debido a que las pérdidas aerodinámicas son proporcionales al área proyectada se necesitan la cuerda de pala en el caso de *flap* y el espesor de pala en el caso de *edge*. En la figura 4 (b) se ilustra cuales serían la cuerda y el espesor en una sección transversal de la pala. La ecuación que describe las pérdidas aerodinámicas sería:

$$F_a = \frac{1}{2} \cdot \rho \cdot C_x \cdot A_x \cdot v^2 \,,$$

donde:

- F_a es la fuerza aerodinámica.
- C_x es el coeficiente aerodinámico. Tomaremos C_x = 1, como si de una placa plana se tratase.
- v es la velocidad de la pala respecto al medio.
- ρ es la densidad del aire. En este caso se tomará la estándar (1.255 kg/m^3).
- A_x es el área proyectada en la dirección del movimiento.

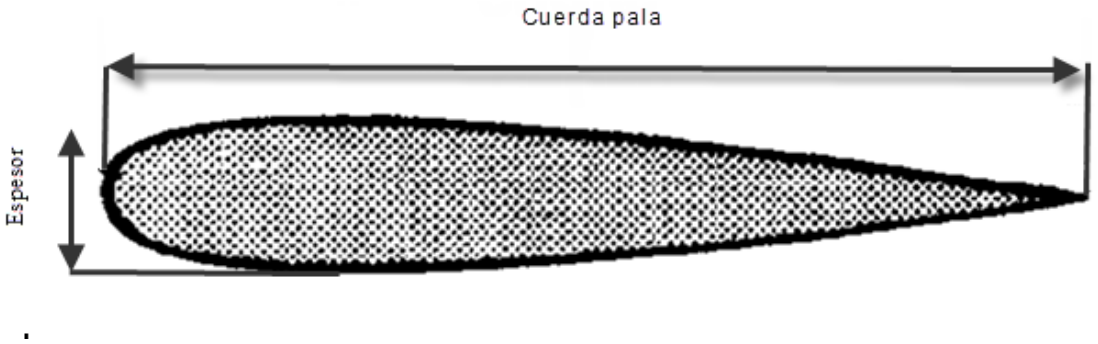


Fig. 4 (b) : Cuerda y espesor de pala.

Las pérdidas aerodinámicas se incluirán en el vector de fuerzas. Para ello deberemos añadir esta componente en cada paso temporal, dado que la velocidad de la pala varía con el tiempo.

Cabe reseñar que todos estos datos se introducirán al programa cargando un fichero Excel, mediante el comando *xlsread* de Matlab. Este fichero Excel tendrá los datos organizados en las seis primeras columnas según la numeración mostrada al inicio del apartado.

4.3 ACTUADORES Y MASAS MUERTAS

En este bloque del programa el usuario debe introducir los datos referentes a los actuadores y masas muertas que vayan a intervenir en la simulación del ensayo a fatiga.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Para ello, la interfaz muestra dos tablas. En la tabla de la izquierda se introduce el radio en el que se va a colocar cada actuador, la masa total y la masa oscilatoria correspondiente a cada uno de los actuadores. En la masa total de los actuadores debe estar incluida la masa oscilatoria de los mismos. Se ha de tener en cuenta que la fuerza ejercida por los actuadores es proporcional a la masa oscilatoria y de naturaleza senoidal. Por lo que, a partir de ella y de la frecuencia de ensayo se calculará la fuerza ejercida por el actuador. El número máximo de actuadores está limitado por 10, cifra muy superior al número empleado en este tipo de ensayos en la actualidad.

La tabla de la derecha es la referente a las masas muertas. En ella se introducen los radios en que están posicionadas las masas muertas y sus respectivas masas. El número máximo de éstas también está limitado por 10. Al igual que en el caso de los actuadores, en la actualidad el número de masas muertas rara vez es superior a cinco.

Obviamente, no disponemos de las propiedades de la pala en un número infinito de puntos, sino que contamos con datos de la pala en una serie de secciones discretas. Por ello, al introducir los actuadores y las masas muertas, es posible que no se encuentren en posiciones en las que las propiedades de la pala son conocidas. Por tanto, en este bloque se realiza una interpolación de los datos conocidos de la pala con el fin de disponer de ciertas aproximaciones de las mismas en los puntos donde el usuario ha elegido introducir los actuadores y las masas muertas.

Además de lo anterior se genera un vector de masas concentradas, en el que se guardan las masas puntuales en cada una de las secciones para las que disponemos de propiedades. Esto nos permitirá, en bloques posteriores calcular la matriz de masas y el vector de fuerzas.

4.4 ANÁLISIS MODAL

Para el bloque de análisis modal se le pide al usuario que introduzca el número de elementos n, con que quiere mallar la pala. Además, el usuario debe decidir si va a simular un ensayo en la dirección de *flap* o de *edge*.

Recordamos que nuestro problema viene modelizado por la ecuación (3.8) deducida anteriormente, que recibe el nombre de la ecuación biarmónica de ondas:

$$\frac{\partial^2}{\partial x^2}\left(D(x)\cdot\frac{\partial^2 u(x,t)}{\partial x^2}\right)+m(x)\cdot\frac{\partial^2 u(x,t)}{\partial t^2}=f(x,t). \quad (4.1)$$

El problema queda completamente definido una vez que se le añaden a la ecuación anterior ciertas condiciones iniciales y de contorno, detalladas más adelante.

En este tercer bloque es donde se discretiza en espacio la ecuación en derivadas parciales (4.1). De dicha discretización se obtienen las matrices de rigidez y de masa, así como el vector de fuerzas. Es decir, pasamos de una ecuación en derivadas parciales cuya incógnita depende de la posición y del tiempo a un sistema de ecuaciones diferenciales cuya incógnita sólo depende del tiempo. En concreto, el sistema semidiscreto resultante tendrá la siguiente forma:

$$[K]\cdot\{u(t)\}+[M]\cdot\frac{d^2\{u(t)\}}{dt^2}=\{f(t)\}\ ,$$

donde:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- $[K] \in \mathbb{R}^{(2n+2)x(2n+2)}$ es la matriz de rigidez global.

- $\{u(t)\} \in \mathbb{R}^{(2n+2)}$ es el vector que aproxima los desplazamientos y su primera derivada, en cada uno de los nodos.

- $[M] \in \mathbb{R}^{(2n+2)x(2n+2)}$ es la matriz de masa global.

- $\{f(t)\} \in \mathbb{R}^{(2n+2)}$ es el vector que aproxima las fuerzas y los momentos aplicados en los nodos.

- Este sistema de ecuaciones diferenciales estaría completado con la discretización de las condiciones iniciales del problema original. Estas se detallan en el apartado de la discretización temporal.

El método empleado para la discretización es el método elementos finitos, habiéndose empleado elementos de Hermite unidimensionales, como se explicará más adelante.

Para la simulación del ensayo es necesario conocer la frecuencia natural de la pala teniendo en cuenta las masas muertas y los actuadores colocados. Ésta es necesaria porque el ensayo se va a desarrollar excitando la pala alrededor de su primera frecuencia natural, dependiendo la fuerza de los actuadores de dicha frecuencia.

Así, en este bloque, una vez completada la discretización en espacio, se procede al cálculo de la frecuencia natural para el ensayo. Esta frecuencia se calcula mediante la teoría del análisis modal, a partir de los valores propios de la matriz resultante de multiplicar la inversa de la matriz de masas por la matriz de rigidez. La teoría del análisis modal empleada se explica más adelante.

Antes de continuar, en este bloque se da como salida la frecuencia natural calculada tanto en hercios como en radianes. Así en función de ésta, el usuario puede decidir el tamaño de paso en el siguiente bloque. Se debe tener en cuenta que la frecuencia natural calculada no es la de la pala, sino la de ésta con las masas muertas y los actuadores. Conocer la frecuencia natural es también importante ya que optimizaremos el ensayo variando la frecuencia por debajo de ésta.

4.5 SIMULACIÓN

En este último bloque se procede a la discretización temporal del problema. Para ello, el usuario debe introducir el paso en el tiempo que crea oportuno. Para la elección del mismo debiera tenerse en cuenta la frecuencia natural calculada en el bloque “Análisis Modal”.

Teniendo en cuenta que la deformada de la pala y, por tanto, su distribución de momentos, se ajusta al modo de vibración de la pala, variando la frecuencia de ensayo podremos ir ajustando la distribución de momentos. Por tanto, también se deberá introducir a que porcentaje de la frecuencia natural se quiere desarrollar el ensayo. Es decir si la frecuencia natural es de un hercio y queremos simular el ensayo con una frecuencia de 0.95 Hz, introduciremos en la casilla indicada 95.

Además el usuario debe cargar un fichero Excel con la distribución de momentos de rango objetivo. Este fichero se carga con el comando *xlsread* de Matlab. Esta distribución es la que queremos alcanzar en la simulación, por la normativa de aerogeneradores referente a ensayos a fatiga *full scale* de palas (ver [15] y [20]).

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Además de elegir el paso temporal, al usuario se le presentan dos opciones:

- Resolver: Resuelve el problema según las masas oscilatorias de los actuadores introducidas en el bloque "Actuadores y Masas Muertas" y la frecuencia de ensayo seleccionada en el presente bloque.

- Optimizar: Optimiza la frecuencia de ensayo para ajustar la distribución de momentos de ensayo (momentos de rango) a la distribución de momentos objetivo introducida por el usuario.

En caso de elegir la opción "Resolver", el programa resuelve el problema temporal para las condiciones introducidas por el usuario. Para la integración temporal emplearemos un esquema Runge-Kutta-Nyström de orden 2, que se describirá más adelante. Al aplicar este método, obtenemos aproximaciones del vector desplazamientos {u} de la pala en diferentes instantes del tiempo.

Denotar que se denominará al {u} vector desplazamientos, si bien incluye también sus primeras derivadas (los giros).

Recordar que el objetivo de este programa es hallar la distribución de momentos de rango a lo largo de la pala, para asegurar que durante el ensayo se le está causando a ésta un determinado daño. Por lo tanto, una vez hallados los desplazamientos se obtiene a partir de los mismos, mediante una doble derivación numérica, la distribución de momentos.

$$\frac{d^2(u)}{d^2(x)} = \frac{M}{EI} \rightarrow \frac{d^2(u)}{d^2(x)} = \frac{M}{D}$$

$$M = \frac{d^2(u)}{d^2(x)} \cdot D$$

Para esta doble derivación numérica se ha empleado el método de las diferencias centrales. Éste se muestra a continuación:

$$M_i = \frac{u_{2i-3} - 2u_{2i-1} + u_{2i+1}}{L^2} \cdot \frac{(D_{i-1} + D_i)}{2} \ .$$

La ecuación anterior se emplea tanto para obtener los momentos de rango a partir de los desplazamientos de rango, como para obtener los momentos medios a partir de los desplazamientos medios.

Una vez tenemos la distribución de momentos de rango, mediante el comando *xlswrite* de Matlab, se da como salida una Excel con un listado de los desplazamientos y de los momentos de rango. En dicho fichero deberán mostrarse también la distribución de momentos medios, que se calcula a partir de los desplazamientos medios, siendo estos últimos obtenidos mediante una simple media de los máximos y los mínimos.

$$u_{m,i} = \frac{u_{max,i} + u_{min,i}}{2} \ ,$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

donde:

- $u_{m,i}$ es el desplazamiento medio en el nodo *i*.
- $u_{max,i}$ es el desplazamiento máximo en el nodo *i*.
- $u_{min,i}$ es el desplazamiento mínimo en el nodo *i*.

Esta media se obtiene para cada nodo al igual que la distribución de momentos de rango. Si bien la salida Excel no se da para todos los nodos, sino solo para las secciones en las que hemos introducido propiedades en el paso "Propiedades de pala". Los desplazamientos y momentos en las secciones para las que teníamos datos, se obtienen por una simple interpolación a partir de los desplazamientos y momentos en los nodos.

También se genera como salida una gráfica con la distribución de momentos obtenida frente a la distribución de momentos objetivo a lo largo de toda la pala. Esta gráfica nos ilustrará lo que nos hemos acercado en el ensayo a los valores que queremos alcanzar. Un ejemplo de dicha gráfica se muestra a continuación.

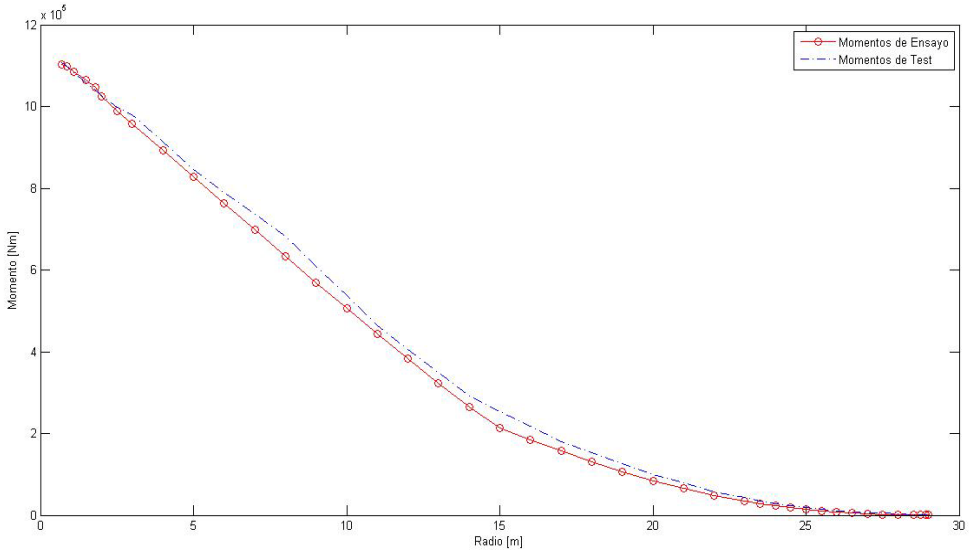


Fig. 4 (c): Comparativa gráfica entre momentos objetivo y momentos de ensayo.

Por otra parte, si se escoge la opción "Optimizar", el programa optimizará la frecuencia de ensayo introducida por el usuario para ajustar al máximo la distribución de momentos de ensayo a la de momentos objetivo. Para ello tomará como iteración inicial los datos introducidos por el usuario. Por ello, será más eficiente elegir unos datos de entrada, tanto de masa oscilatoria como de frecuencia de ensayo, razonables.

Una vez resuelta la iteración inicial y comparadas ambas distribuciones de momentos. Se procede a variar la frecuencia de ensayo. En caso de que la resta del área de momentos objetivo menos el área de momentos de ensayo sea positiva, se aumentará la frecuencia. Habrá que estar siempre por debajo de la frecuencia natural calculada, ya que en teoría si excitamos la pala en la frecuencia natural las

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

vibraciones se amplificarían infinitamente. En el caso de que la resta mencionada anteriormente fuese negativa se disminuiría la frecuencia. Este proceso se repetirá hasta que la diferencia tenga un error inferior al 3 % o hasta que se vaya a alcanzar la frecuencia natural.

Las salidas deberán ser las mismas que en el caso de “Resolver”, mostrando únicamente los de la iteración final. Con esta opción se obtendrá la frecuencia óptima de ensayo. En el caso en el que con esta opción quedara alguna zona de la pala sobrecargada, el usuario debería hacer el ajuste final manualmente con la opción “Resolver”.

Si en este ajuste final solamente se quiere variar la frecuencia de ensayo, se puede hacer variando desde el último bloque el porcentaje de la frecuencia de ensayo. Sin embargo en caso de que se vea que con esas masas muertas y esos actuadores no se puede alcanzar una distribución lo suficientemente buena, habrá que volver al punto “Actuadores y Masas Muertas” y modificar adecuadamente dichos datos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

5 RESOLUCIÓN DE LA ECUACIÓN BIARMÓNICA DE ONDA

5.1 DISCRETIZACIÓN EN ESPACIO

5.1.1 ELEMENTOS BARRA UNIDIMENSIONALES

El problema a resolver es unidimensional, ya que tanto las fuerzas como los desplazamientos ocurren en una misma dimensión. En consecuencia resulta innecesario coger elementos más complejos, dado que el resto de dimensiones no aportan información adicional. Por lo tanto emplearemos elementos barra unidimesionales.

5.1.2 FORMULACION VARIACIONAL

Para obtener la formulación variacional de nuestro problema partimos de la ecuación inicial deducida anteriormente. Esta ecuación es empleada en diversa bibliografía como por ejemplo en [2], [3] o [14]. Recibe el nombre de ecuación biarmónica de ondas y se muestra a continuación:

$$\frac{\partial^2}{\partial x^2}\left(D(x)\cdot\frac{\partial^2 u(x,t)}{\partial x^2}\right)+m(x)\cdot\frac{\partial^2 u(x,t)}{\partial t^2}=f(x,t)$$

con x ∈ (0,L), siendo L la longitud de la pala y $t \in (0,T]$, además de las apropiadas condiciones de contorno e iniciales.

Para deducir las condiciones de contorno de nuestro problema es importante conocer la ecuación de la elástica según la teoría de Navier-Bernoulli:

$$\frac{\partial^2 u}{\partial^2 x}=\frac{M}{EI}\rightarrow\frac{\partial^2 u}{\partial x^2}=\frac{M}{D}\ . \quad (5.1)$$

De la expresión anterior se extrae que la segunda derivada del desplazamiento respecto al espacio es el momento dividido por la rigidez a flexión de la viga. Además, por la resistencia de materiales se puede deducir que el cortante es la derivada del momento, como se demuestra a continuación a partir de un simple equilibrio estático de una sección diferencial de viga.

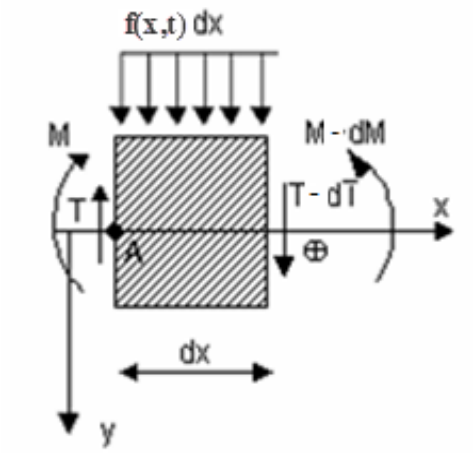


Fig. 5 (a): Sección transversal de una viga

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Al estar la viga en equilibrio, la suma de fuerzas verticales y la suma de momentos respecto al punto A son cero, es decir:

$$\sum F_v = 0 \ \rightarrow \ -T + T - dT + q \cdot dx = 0 \ \rightarrow \frac{dT}{dx} = q \ ,$$

$$\sum M_A = 0 \ \rightarrow M - dM) + q \cdot dx \cdot \frac{dx}{2} - M + (T + dT) \cdot dx = 0 \ \rightarrow \frac{dM}{dx} = T \ ,$$

donde:

- M [Nm] es el momento.
- q [N/m] es la carga o fuerza por unidad de longitud.
- T [N] es el cortante.
- $\sum F_v$ [N] es el sumatorio de fuerzas verticales.
- $\sum M_A$ [Nm] es el sumatorio de momentos respecto al punto A.

La condición de empotramiento implica por definición que el desplazamiento y el giro son nulos en el extremo empotrado. En el caso de extremo libre, haciendo también por definición el momento y el cortante son nulos, puesto que si no lo fuera así no habría nada que los compensase y ese extremo dejaría de estar en equilibrio.

A partir de las igualdades anteriores y de la ecuación (5.1) se deducen fácilmente las siguientes condiciones de contorno para nuestra viga, que se encuentra empotrada en un extremo (x=0) y libre en el otro (x=L).

- $u(0) = 0$: el extremo empotrado no se desplaza.
- $\frac{\partial u(0)}{\partial x} = 0$: el extremo empotrado no gira.
- $\frac{\partial^2 u}{\partial x^2}(L) = 0$: el momento en el extremo libre es igual a cero (punta de pala).
- $T(L) = \frac{\partial}{\partial x}\left(D \cdot \frac{\partial^2 u}{\partial x^2}(L)\right) = \frac{\partial D}{\partial x} \cdot \frac{\partial^2 u}{\partial x^2}(L) + D \cdot \frac{\partial^3 u}{\partial x^3}(L) = 0 \xRightarrow[D \neq 0 \wedge \frac{\partial^2 u}{\partial x^2}(L)]{} \frac{\partial^3 u}{\partial x^3}(L) = 0$: el cortante es cero en el extremo libre (punta de pala).
- A continuación, procedemos a deducir la formulación variacional de nuestro problema. Consideramos el espacio:
-
  - $H^2(0, L) \coloneqq \{u \in L^2(0, L) | u', u'' \in L^2(0, L)\}$,
  -
- y su subespacio:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

-
- $V \coloneqq \{u \in H^2(0,L) |\, u(0) = u'(0) = 0\}$.

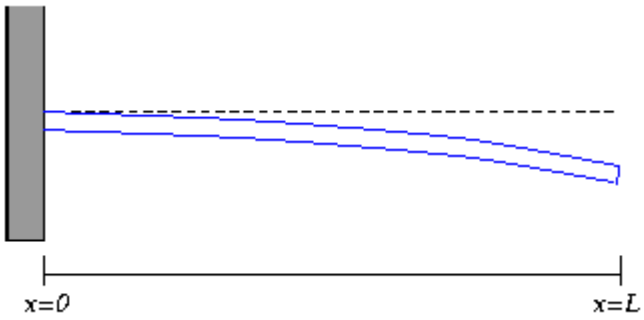


Fig. 5 (b): Viga empotrada.

Partiendo de la ecuación original:

$$\frac{\partial^2}{\partial x^2}\left(D(x)\cdot\frac{\partial^2 u(x,t)}{\partial x^2}\right)+m(x)\cdot\frac{\partial^2 u(x)}{\partial t^2}=f(x,t)\,,$$

integramos en el dominio multiplicando por una función “v”, perteneciente al subespacio V:

$$\int_0^L \frac{\partial^2}{\partial x^2}\left(D(x)\cdot\frac{\partial^2 u(x,t)}{\partial x^2}\right)\cdot \mathrm{v}\cdot \mathrm{dx}+\int_0^{\mathrm{L}}\left(m(x)\cdot\frac{\partial^2 u(x,t)}{\partial t^2}\right)\cdot \mathrm{v}\cdot \mathrm{dx}=\int_0^{\mathrm{L}} f(x,t)\cdot \mathrm{v}\cdot \mathrm{dx}$$

Aplicando el teorema de Green así como las condiciones de contorno consideradas llegamos a la formulación variacional de nuestro problema:

$$\int_0^L\left(D(x)\cdot\frac{\partial^2 u(x,t)}{\partial x^2}\cdot\frac{\partial^2 v(x)}{\partial x^2}\right)\cdot \mathrm{dx}+\int_0^{\mathrm{l}} m(x)\cdot\frac{\partial^2 u(x,t)}{\partial x^2}\cdot \mathrm{v}\cdot \mathrm{dx}=\int_0^{\mathrm{L}} f(x,t)\cdot \mathrm{v}\cdot \mathrm{dx}\ ,$$

donde asumimos que $\forall t\ u(x,t) \in V\ \forall t$

Como ya habíamos avanzado, para modelizar el problema de la viga de Euler-Bernoulli es necesario emplear elementos de Hermite. En [7] se detalla el por qué de ese requerimiento.

5.1.3 ELEMENTOS HERMITE P3

Las Hermitianas cúbicas consisten en dos tipos de funciones definidas a lo largo de cada elemento. Un tipo está asociado a la interpolación de la solución aproximada de los desplazamientos y el otro a su derivada.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Por cada elemento aparecen dos hermitianas de tipo uno. Habiendo una función completa de este tipo por cada nodo, que valdrá uno en el nodo para el que está definida y cero en los nodos adyacentes.

Las hermitianas cúbicas de tipo dos interpolan las derivadas en los nodos, valiendo éstas cero en el nodo para el que están definidas y uno su derivada en dicho nodo. En los nodos adyacentes vale cero tanto la función como su derivada. También en este caso coexisten dos por elemento, una por cada nodo.

Para una integración más fácil a lo largo de cada elemento, haremos un cambio a coordenadas locales, definiendo estas como $\xi = -1$ en el nodo inicial del elemento y $\xi = 1$ en el nodo final. Por tanto las coordenadas locales de cada elemento irán de -1 a1.

A continuación, se definen en coordenadas locales las cuatro hermitianas cúbicas que coexisten en cada elemento, también llamadas funciones de forma, cuyas gráficas se muestran en la figura 5 (c):

$$N_{u1}^{e} = \frac{1}{4} \cdot (\xi - 1)^2 \cdot (\xi + 2)$$

$$N_{u2}^{e} = -\frac{1}{4} \cdot (\xi + 1)^2 \cdot (\xi - 2)$$

$$N_{\theta 1}^{e} = \frac{1}{8} \cdot L^{e} \cdot (\xi + 1) \cdot (\xi - 1)^2 \ ,$$

$$N_{\theta 2}^{e} = \frac{1}{8} \cdot L^{e} \cdot (\xi + 1)^2 \cdot (\xi - 1) \ .$$

En el cambio de variable se admite la siguiente expresión:

$$\xi = 2 \cdot \left[\frac{(x - x_1^e)}{L^e} \right] - 1 \ ,$$

$$L^e = x_2^e - x_1^e \ ,$$

$$\frac{dx}{d\xi} = \frac{1}{2} \cdot L^e \quad \text{con} \quad \xi \in [-1,1] \ ,$$

donde:

- x es la coordenada longitudinal global de la pala.
- ξ es la coordenada longitudinal local del elemento.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- x_1^e es la coordenada global x del nodo inicial del elemento.
- x_2^e es la coordenada global x del nodo final del elemento.
- L^e es la longitud del elemento. A partir de este momento consideramos que todos los elementos tienen la misma longitud, a la que denotaremos por L.

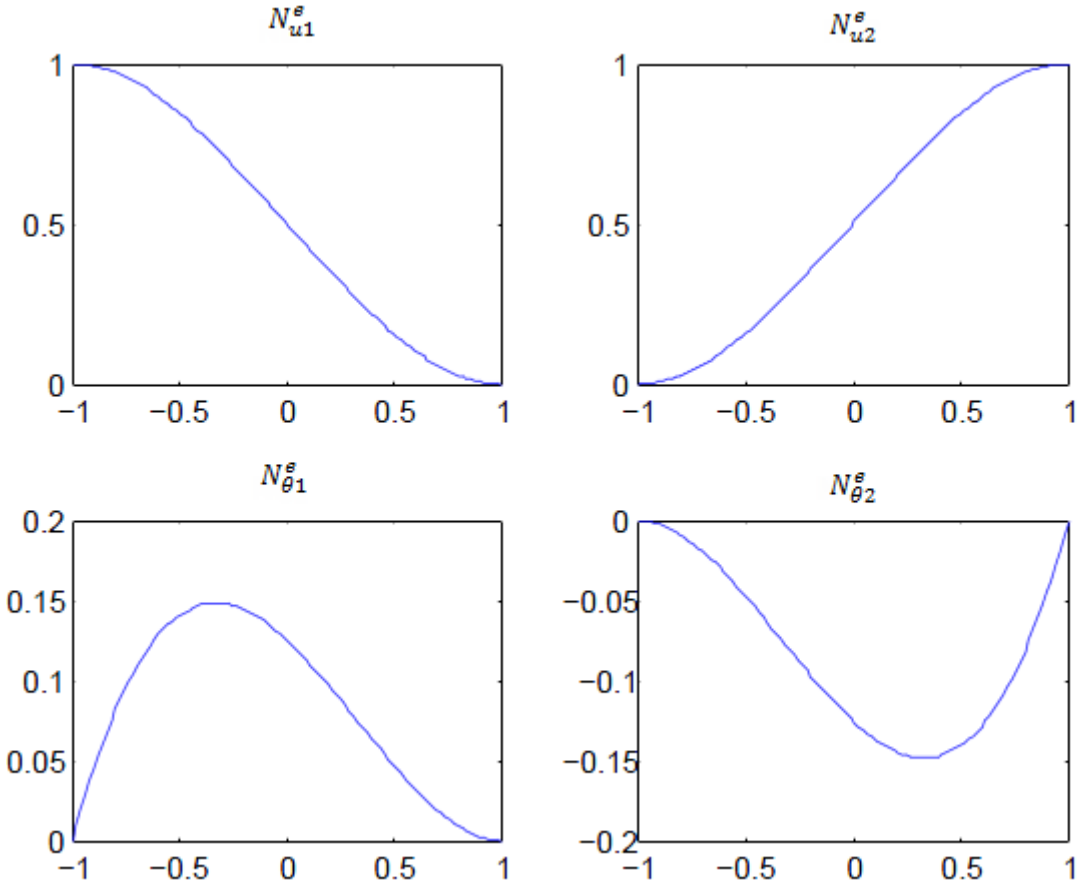


Fig. 5 (c): Funciones de forma en coordenadas locales para el elemento e.

Por tanto la función de interpolación del vector desplazamientos para el elemento e en coordenadas locales queda de la siguiente manera:

$$\{\hat{u}^e\} = \{\hat{u}(x,t)|_e\} = \{N^e(x)\} \cdot \{u^e(t)\} \ ,$$

$$\{\hat{u}^e\} = [N_{u1}^e(x) \quad N_{\theta 1}^e(x) \quad N_{u2}^e(x) \quad N_{\theta 2}^e(x)] \cdot \begin{bmatrix} u1(t) \\ \theta 1(t) \\ u2(t) \\ \theta 2(t) \end{bmatrix} =$$

$$= u_1^e(t) \cdot N_{u1}^e(x) + u_2^e(t) \cdot N_{u2}^e(x) + \left(\frac{du_1^e}{dx}\right)(t) \cdot N_{\theta 1}^e(x) + \left(\frac{du_2^e}{dx}\right)(x) \cdot N_{\theta 2}^e(x) \ .$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

5.1.4 MATRIZ DE RIGIDEZ

Para obtener la matriz de rigidez sustituimos en la expresión

$$\int_0^L \left(D(x) \cdot \frac{\partial^2 u(x,t)}{\partial x^2} \cdot \frac{\partial^2 v(x)}{\partial x^2} \right) \cdot dx \ ,$$

la función de desplazamientos por su aproximación $\hat{u}$ obteniendo:

$$\int_0^L \left(D(x) \cdot \frac{\partial^2 \hat{u}(x,t)}{\partial x^2} \cdot \frac{\partial^2 v(x)}{\partial x^2} \right) \cdot \mathrm{dx} \ .$$

Por la aditividad de la integral respecto del intervalo se tiene:

$$\int_0^L \left(D(x) \cdot \frac{\partial^2 \hat{u}(x,t)}{\partial x^2} \cdot \frac{\partial^2 v(x)}{\partial x^2} \right) \cdot \mathrm{dx} = \sum_{\mathrm{e}} \int \left(D(x) \cdot \frac{\partial^2 \hat{u}(x,t)}{\partial x^2} \cdot \frac{\partial^2 v(x)}{\partial x^2} \right) \cdot \mathrm{dx} \ ,$$

donde e denota a cada uno de los elementos en los que hemos dividido la viga.

Por último, haciendo un cambio de variable a coordenadas locales en la integral asociada a cada elemento, obtenemos la matriz de rigidez del mismo. Una vez hecho esto bastará con ensamblar las matrices de rigidez de los distintos elementos para obtener la matriz de rigidez global.

Notar que de aquí en adelante denominaremos vector desplazamientos al vector {u}, a pesar de que además de los desplazamientos incluye sus derivadas.

Las segundas derivadas para la función desplazamientos en el elemento e tinen la forma:

$$\frac{\partial^2 \{\hat{u}^e(x,t)\}}{\partial x^2} = \frac{4}{L^2} \cdot \frac{\partial^2 \{\hat{u}^e(\xi,t)\}}{\partial \xi^2} = \frac{4}{L^2} \cdot \frac{\partial^2 \{N^e(\xi)\}}{\partial \xi^2} \cdot \{u^e(t)\} = \{B\} \cdot \{u^e(t)\} \ ,$$

donde:

$$\{B\} = \frac{1}{L} \cdot \left[6 \cdot \frac{\xi}{L} \quad 3\xi - 1 \quad -6\frac{\xi}{L} \quad 3\xi + 1 \right] \ .$$

A partir de aquí, la matriz de rigidez del elemento se puede calcular como:

$$K^e = \int_{-1}^{1} D^e \cdot \{B\}^T \cdot \{B\} \cdot \frac{1}{2} \cdot L \cdot d\xi \ .$$

donde D^e es la rigidez de la viga en el elemento e.

Basta con integrar unos pocos elementos para darse cuenta de que las matrices de rigidez para todos ellos son iguales, siendo esto debido la uniformidad de los mismos. A continuación se muestra la matriz de rigidez de un elemento:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

$$[K^e] = D^e \begin{bmatrix} \frac{12}{L^3} & \frac{6}{L^2} & \frac{-12}{L^3} & \frac{6}{L^2} \\ \frac{6}{L^2} & \frac{4}{L} & \frac{-6}{L^2} & \frac{2}{L} \\ \frac{-12}{L^3} & \frac{-6}{L^2} & \frac{12}{L^3} & \frac{-6}{L^2} \\ \frac{6}{L^2} & \frac{2}{L} & \frac{-6}{L^2} & \frac{4}{L} \end{bmatrix}$$

donde:

- L es la longitud del elemento, que en el caso que nos concierne es constante a lo largo de toda la viga (pala).
- D^e es la rigidez del elemento que variará de un elemento a otro, dado que la viga no es uniforme.

Ensamblar la matriz de rigidez consiste en unir las matrices de rigidez, sumando las componentes de los grados de libertad comunes. A modo explicativo, se muestra a continuación la matriz de rigidez global de una viga de dos elementos:

$$[K] = \begin{bmatrix} D^1\frac{12}{L^3} & D^1\frac{6}{L^2} & D^1\frac{-12}{L^3} & D^1\frac{6}{L^2} & 0 & 0 \\ D^1\frac{6}{L^2} & D^1\frac{4}{L} & D^1\frac{-6}{L^2} & D^1\frac{2}{L} & 0 & 0 \\ D^1\frac{-12}{L^3} & D^1\frac{-6}{L^2} & D^1\frac{12}{L^3} + D^2\frac{12}{L^3} & D^2\frac{6}{L^2} + D^1\frac{-6}{L^2} & D^2\frac{-12}{L^3} & D^2\frac{6}{L^2} \\ D^1\frac{6}{L^2} & D^1\frac{2}{L} & D^2\frac{6}{L^2} + D^1\frac{-6}{L^2} & D^2\frac{4}{L} + D^1\frac{4}{L} & D^2\frac{-6}{L^2} & D^2\frac{2}{L} \\ 0 & 0 & D^2\frac{-12}{L^3} & D^2\frac{-6}{L^2} & D^2\frac{12}{L^3} & D^2\frac{-6}{L^2} \\ 0 & 0 & D^2\frac{6}{L^2} & D^2\frac{2}{L} & D^2\frac{-6}{L^2} & D^2\frac{4}{L} \end{bmatrix}$$

En la figura 5 (d) se muestra la distribución de elementos no nulos en la matriz de rigidez global de una viga de cuatro elementos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

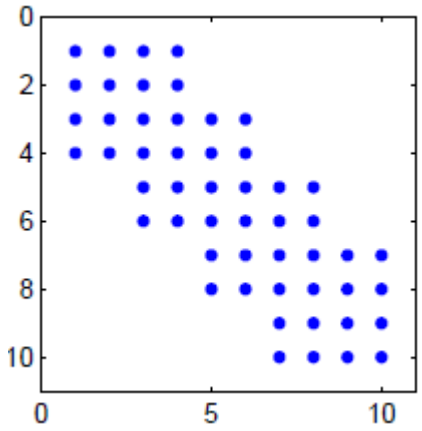


Fig. 5 (d): Matriz de rigidez global para 4 elementos.

Como puede observarse a medida que el número de elementos aumenta, la matriz se vuelve cada vez más *sparse*, es decir va aumentando la proporción de elementos nulos frente a los no nulos. Solamente van quedando elementos distintos de cero cerca de la diagonal. Esto es característico de matrices estructurales tanto de masa como de rigidez y deberá tenerse en cuenta a la hora de resolver en Matlab los sistemas de ecuaciones lineales resultantes. Para ello, utilizaremos los comandos para matrices *sparse*, que evitan almacenar en memoria todos los elementos nulos de la matriz, guardando simplemente aquellos elementos que me aportan información.

5.1.5 MATRIZ DE MASA

Para cálculos dinámicos con elementos finitos no es suficiente con la matriz de rigidez, sino que hace falta también la matriz de masa.

La conformación de la matriz de masas global es similar a la de la matriz de rigidez. Primero se integran en coordenadas locales las matrices de rigidez de cada uno de los elementos y posteriormente se pasan a coordenadas globales para ensamblar la matriz de masas global, con las mismas técnicas empleadas para la matriz de rigidez.

La matriz de masa debe cumplir las siguientes características:

- *Simetría*: $[\mathrm{M}]^{\mathrm{T}} = [\mathrm{M}^{\mathrm{e}}]$.
- *Simetría Física*: las simetrías de los elementos deben de mostrarse en la matriz de masas.
- *Conservación*: las masas translacionales nodales deben ser las mismas que las de la barra.
- *Definición positiva*: $[\mathrm{M}^{\mathrm{e}}]$ debe tener valores propios positivos.

Por lo tanto, el primer paso e integrar elemento a elemento la siguiente ecuación, realizando para ello un cambio a coordenadas locales:

$$\int_0^{\mathrm{L}} \left(m(x) \cdot \frac{\partial^2 \hat{u}}{\partial t^2} \right) \cdot \mathrm{v} \cdot \mathrm{dx} \quad .$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Recordar, que según el teorema de Galerkin tomaremos “v” como el vector de funciones de forma.

Al final la integral que se debe llevar a cabo sería:

$$[M^e] = \int_{-1}^{1} m^e \cdot \frac{L}{2} \cdot \{N^e\}^T \cdot \{N^e\} \cdot d\xi \quad ,$$

donde, al ser la masa constante en cada elemento, ésta puede salir fuera de la integral.

$$[M^e] = m^e \cdot \int_{-1}^{1} \frac{L}{2} \cdot \{N^e\}^T \cdot \{N^e\} \cdot d\xi \quad .$$

siendo $\{N^e\}$ el vector de funciones de forma en coordenadas locales.

Tras integrar en coordenadas locales la matriz de masa de un elemento queda como se muestra:

$$[M^e] = \frac{m^e \cdot L}{420} \cdot \begin{bmatrix} 156 & 22 \cdot L & 54 & -13 \cdot L \\ 22 \cdot L & 4 \cdot L^2 & 13 \cdot L & -3 \cdot L^2 \\ 54 & 13 \cdot L & 156 & -22 \cdot L \\ -13 \cdot L & -3 \cdot L^2 & -22 \cdot L & 4 \cdot L^2 \end{bmatrix}$$

El proceso para ensamblar la matriz de masas global es exactamente igual que para ensamblar la matriz de rigidez global.

Dado que se va a necesitar invertir la matriz de masa, sería muy útil aplicar la técnica de *mass lumping*, según la cual la matriz de masas es aproximada por una matriz diagonal. A la matriz de masa diagonalizada se le conoce como matriz de masas concentradas y su inversión es inmediata.

Esta matriz de masas concentradas obviamente debe tener la misma masa en los nodos translacionales que la masa de la barra. Sin embargo no hay consenso para las masas rotacionales. A continuación se muestra la matriz de masa concentrada, con las masas rotacionales en función de un parámetro α.

$$[M^e] = m^e \cdot L \cdot \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \alpha \cdot L^2 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \alpha \cdot L^2 \end{bmatrix}$$

Aquí α es un parámetro no negativo, usualmente tomado entre 0 y 1/50. La elección de α ha sido un tema extensamente discutido durante décadas en la bibliografía de elementos finitos (ver [1] y [2]). Aun así todas esas discusiones han sido en vano, pues hoy en día sigue sin estar determinado. En teoría el mejor α para cuadrar los momentos angulares sería 0, pero este valor da una matriz singular. Una matriz singular es indeseable en aquellas situaciones en las que ésta aparezca invertida, como es el caso que nos concierne.

En nuestro caso particular, se ha escogido α = 1/420. Para verificar que la elección ha sido correcta lo que se ha hecho ha sido calcular la frecuencia natural de varias palas. Se ha optado por esta comprobación porque la frecuencia natural se obtiene a partir de los valores propios de $[M]^{-1} \cdot [K]$. Las diferencias obtenidas entre emplear la matriz de masas concentradas y emplear la real han sido

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

siempre inferiores al 0.5 %. Concretamente calculando la frecuencia natural en *edge* de una pala de Gamesa se obtuvieron los siguientes resultados:

- Matriz de masas concentradas → 1.6481 Hz
- Matriz de masas real → 1.6487 Hz

Por tanto la matriz de masas concentradas de cada elemento empleada queda como sigue a continuación:

$$[M^e] = m^e \cdot L \cdot \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{L^2}{60} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{L^2}{60} \end{bmatrix}$$

5.1.6 VECTOR DE FUERZA

Por último, una vez obtenida la matriz de rigidez y la de masa, solamente queda discretizar el vector de fuerza. Este proceso es similar al empleado para discretizar las matrices de masa y rigidez.

El vector de fuerzas se obtendrá de integrar en el dominio la parte derecha de la igualdad de la ecuación inicial.

$$\int_0^{\mathrm{L}} f(x,t) \cdot \mathrm{v} \cdot \mathrm{dx}$$

Nuevamente descomponiendo la anterior integral en integrales sobre cada elemento y considerando que la función "v" es el vector de funciones de forma $\{N^e\}$, el vector de fuerza para carga uniforme en el elemento e queda de la siguiente manera:

$$\{f^e\} = \int \{N(x)\}^T \cdot q^e(x,t) \cdot dx = \int_{-1}^{1} \{N(\xi)\}^T \cdot q^e(\xi,t) \cdot \frac{1}{2} \cdot L \cdot d\xi$$

siendo q^e la carga por unidad de longitud en el elemento e, a lo largo del cual es constante

Esto da lugar a un vector de fuerzas para cada elemento:

$$\{f^e\} = \frac{1}{2} \cdot q^e \cdot L \cdot \int_{-1}^{1} \{N\}^T \cdot d\xi = \frac{1}{2} \cdot q^e \cdot L \cdot \int_{-1}^{1} \begin{bmatrix} \frac{1}{4} \cdot (\xi-1)^2 \cdot (\xi+2) \\ \frac{1}{8} \cdot L^e \cdot (\xi+1) \cdot (\xi-1)^2 \\ -\frac{1}{4} \cdot (\xi+1)^2 \cdot (\xi-2) \\ \frac{1}{8} \cdot L^e \cdot (\xi+1)^2 \cdot (\xi-1) \end{bmatrix} \cdot d\xi = \{q^e\} \cdot L \cdot \begin{bmatrix} \frac{1}{2} \\ \frac{L}{12} \\ \frac{1}{2} \\ -\frac{L}{12} \end{bmatrix}$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Este sería el vector de fuerzas para un elemento barra. Como se puede ver la carga uniforme q^e da lugar a dos fuerzas en los nodos, $\frac{q^e \cdot L}{2}$, más dos momentos en los extremos, $\frac{q^e \cdot L^2}{12}$. Estos momentos son los de empotramiento en los extremos de la barra.

De lo anterior podemos concluir que en las posiciones impares del vector de fuerzas se colocan las fuerzas aplicadas en los nodos correspondientes, mientras que en las posiciones pares se colocan los momentos aplicados en los nodos correspondientes.

Teniendo en cuenta que en nuestro caso no se aplican momentos, salvo en el nodo del empotramiento, las posiciones pares del vector de fuerzas global ensamblado serán todas 0, salvo la posición segunda correspondiente al momento de empotramiento.

5.1.7 ANÁLISIS MODAL

En el tercer punto del programa, "Modal Analysis", se calcula la frecuencia natural de la pala teniendo en cuenta las masas muertas y los actuadores. Para calcular la frecuencia natural de la pala se puede recurrir a diversos métodos:

- Conservación de la energía.
- Dunkerley.
- Análisis modal

El método escogido para calcular la frecuencia natural ha sido el del análisis modal. Entre otras cosas porque, al tener discretizadas las matrices de masa y rigidez, resulta directo obtener los valores propios de la inversa de la matriz de masa por la de rigidez. Como explicaremos ahora, según el análisis modal, las raíces de estos autovalores son las frecuencias naturales..

La frecuencia natural de un sistema es aquella a la que dicho sistema vibraría libremente sin tener en cuenta el amortiguamiento. Por tanto, al ser vibraciones libres no tenemos en cuenta el vector de fuerzas y la ecuación matricial que se nos presenta es la siguiente.

$$[K] \cdot \{u\} + [M] \cdot \{\ddot{u}\} = 0 \qquad (5.2)$$

Resulta evidente que este sistema tiene al menos la solución trivial homogénea. Obviamente, no es ésa la solución que buscamos dado que para esa solución los desplazamientos (u) y las aceleraciones (ü), son cero. Es decir, para la solución homogénea no hay movimiento, por tanto, no hay vibración.

Visto que la solución de vibraciones libres no es la solución homogénea, pasamos a analizar la forma que deben tener las vibraciones:

$$\{u\} = \{U\} \cdot \sin(\omega \cdot t) \qquad (5.3)$$

Donde:

- $\{u\}$: es el vector de desplazamientos y de giros. Como se ha comentado anteriormente nos referiremos a él simplemente como el vector de desplazamientos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

- $\{U\}$: es el vector de amplitudes.

Sustituyendo (5.3) en la ecuación de vibraciones libres (5.2), obtenemos:

$$[K] \cdot \{U\} = \omega^2 \cdot [M] \cdot \{U\}$$

Teniendo en cuenta que [M] es regular, la expresión anterior es equivalente a:

$$[M]^{-1} \cdot [K] \cdot \{U\} = \omega^2 \cdot \{U\} \ ,$$

que constituye un problema de valores propios. En dicho problema, el vector amplitud, $\{\boldsymbol{U}\}$, será el vector propio, también llamado en este caso modo de vibración pues es quien describe cómo vibra la pala. Por otra parte, el valor propio será $\boldsymbol{\omega^2}$. De hecho, como hemos resuelto el problema en que el sistema vibra libremente sin amortiguación, ω será la frecuencia natural de vibración del sistema en radianes.

Así, las frecuencias naturales serán los valores propios de $[M]^{-1} \cdot [K]$. Teniendo en cuenta que el ensayo a fatiga de palas se desarrolla alrededor de la primera frecuencia de vibración, nos interesa esta y no el resto. Al estar buscando el menor valor propio se implementa en Matlabel método de la potencia inversa.

Por otra parte, se observa que el cálculo de los valores propios requiere calcular la inversa de la matriz de masas. En el desarrollo de este programa, como se mencionó anteriormente, estamos trabajando con la matriz de masas concentradas. Esta matriz es diagonal, lo que computacionalmente favorece el cálculo de la matriz inversa; de hecho, $[M]^{-1}$ será una matriz diagonal cuyos elementos diagonales son los inversos de los elementos diagonales de la matriz de masas.

La primera frecuencia de vibración es la que tiene el valor más bajo. Por ello, tomaremos el valor propio más bajo y le aplicamos la raíz cuadrada, obteniendo la primera frecuencia natural de vibración en radianes.

En principio la frecuencia de vibración libre es una combinación lineal de las frecuencias naturales. Sin embargo, se puede excitar la pala con unas determinadas condiciones de contorno de manera que se haga vibrar a la pala según sólo uno de sus modos.

Debido a que puede haber usuarios más familiarizados con hercios que con radianes, en pantalla se mostrará la solución también en hercios.

5.2 DISCRETIZACIÓN EN EL TIEMPO

La discretización espacial explicada anteriormente a dado lugar a un sistema de ecuaciones diferenciales de la forma:

$$\{U''(t)\} = \{F(t, \{U(t)\})\} \ , \qquad (5.4)$$

donde:

$$\{F(t, \{U(t)\})\} = [M]^{-1} \cdot (\{f(t)\} - [K]\{U(t)\}) \ .$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Este sistema de ecuaciones diferenciales viene acompañado de la discretización de las condiciones iniciales del problema original que denotaremos como $\{U_0\}$ y $\{V_0\}$. $\{U_0\}$ será el vector de desplazamientos inicial, que lo tomaremos como la deformada de la pala por su peso propio y el de las masas muertas y actuadores. $\{V_0\}$ será el vector que aproxima a las derivadas primeras del vector deplazamientos respecto del tiempo, es decir, el vector velocidades en el instante inicial. Tomaremos éste como el vector, ya que la pala parte del reposo.

Para la discretización en tiempo vamos a utilizar un método Runge-Kutta-Nyström (RKN) que, aplicado a un sistema de ecuaciones genérico del tipo a (5.4), admite la expresión:

$$Q_{n,i} = U_n + c_i \cdot k \cdot V_n + k^2 \cdot \sum_{i=1}^{s} a_{ij} \cdot F\left(t_n + c_j \cdot k, Q_{n,j}\right) , \qquad i = 1, \dots, s.$$

$$V_{n+1} = V_n + k \cdot \textstyle\sum_{i=1}^{s} b_i \cdot F\left(t_n + c_i \cdot k, Q_{n,i}\right) , \tag{5.5}$$

$$U_{n+1} = U_n + k \cdot V_n + k^2 \cdot \sum_{i=1}^{s} \beta_i \cdot F\left(t_n + c_i \cdot k, Q_{n,i}\right) ,$$

- $(\{U_n\} \quad \{V_n\})^T$ la aproximación numérica a la solución exacta $(\{U(tn)\} \quad \{U'(tn)\})^T$.
- k : es el tamaño de paso en el tiempo.
- t_n es igual a $n \cdot k$.
- a_{ij}, b_i, c_i y β_i son los coeficientes específicos del método RKN empleado.
- s es el número de etapas del método.

Los coeficientes a_{ij}, b_i , c_i y β_i , se suelen almacenar en un tablero, conocido como tablero de Butcher que tiene la forma:

c	A
	β^T
	b^T

Siendo c, β y b, vectores de longitud s y A una matriz de dimensión $s \times s$. Cuando la matriz A es triangular con ceros en la diagonal se le denomina método explícito, mientras que en el resto de los casos recibe el nombre de método implícito.

Las condiciones *c, β, b* y A deben cumplir para que el método RKN alcance orden clásico *p*, al integrar la ecuación diferencial ordinaria lineal no homogénea con coeficientes independientes del tiempo son:

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

$$\beta^T \cdot A^m \cdot c^l = \frac{1}{(l+2m+2)\cdot(l+2m+1)\cdot \ldots \cdot (l+1)} \ , 0 \le l+2m \le p-2 \ , \quad (5.3)$$

$$b^T \cdot A^m \cdot c^l = \frac{1}{(l+2m+1)\cdot(l+2m)\cdot \ldots \cdot) l+1)} \ , \ 0 \le l+2m \le p-1 \ , \quad (5.4)$$

donde $c^l = [c_1^l, \ldots, c_s^l]^T, l \ge 0$ entero.

Propiedades adicionales de estos métodos, como la estabilidad, puede consultarse en bibliografía especializada, como por ejemplo [21].

Para este proyecto el método escogido tiene orden 3, es R-estable y presenta el siguiente tablero (ver [21]):

$\frac{9+\sqrt{33}}{12}$	$\frac{(9+\sqrt{33})^2}{288}$	0
$\frac{1+\sqrt{33}}{6+2\sqrt{33}}$	$-\frac{99+19\sqrt{33}}{4\left(3+\sqrt{33}\right)^2}$	$\frac{(9+\sqrt{33})^2}{288}$
	$\frac{3-\sqrt{33}}{54+6\sqrt{33}}$	$\frac{2\left(6+\sqrt{33}\right)}{3\left(9+\sqrt{33}\right)}$
	$\frac{2}{9+\sqrt{33}}$	$\frac{7+\sqrt{33}}{9+\sqrt{33}}$

5.3 PROCESO DE OPTIMIZACIÓN

Para optimizar la frecuencia de ensayo utilizaremos un método similar al del punto medio para el cálculo de raíces de un polinomio. En los ensayos a fatiga se trabaja justo por debajo de la primera frecuencia natural. En base a la experiencia tomamos un rango de frecuencias entre 0.8 veces y 0.985 veces la frecuencia natural. El primero será la cota inferior y el segundo la cota superior de trabajo.

El usuario debe introducir la frecuencia de ensayo para la primera iteración como un porcentaje de la frecuencia natural de la pala (actuadores y masas muertas incluidos). El porcentaje de la primera iteración debe ser elegido dentro del rango de trabajo, es decir, entre el 80% y el 98.5% de la frecuencia natural.

Una vez se inicia el proceso de optimización lo primero que hace el método es resolver el problema para la frecuencia introducida por el usuario. Cuando lo resuelve, compara los momentos de ensayo con los objetivo y calcula el error relativo de la siguiente manera:

$$Error = \frac{M_{ensayo}}{M_{objetivo}} - 1$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Esta fórmula la aplica para cada sección en la que tenemos datos desde el inicio de la pala hasta donde se encuentra el primer último actuador. Solo compara hasta el último actuador, puesto que solo en las entre el inicio de la pala y estos, podemos cambiar significativamente la distribución de momentos al variar la frecuencia.

Posteriormente se hace la media de esos errores:

- Si es positiva habrá que disminuir la frecuencia.
- Si es negativa habrá que incrementar la frecuencia.

A partir de aquí se inicia el bucle iterativo optimizador, que repite continuamente los siguientes pasos:

1. Cálculo del error medio. Si es negativo se cambia la cota mínima por la frecuencia de la anterior. Si por el contrario el error medio es positivo se cambia la cota superior por la frecuencia de la anterior iteración.
2. Cálculo de la frecuencia para la siguiente iteración como el punto medio de la cota superior y la inferior.
3. Resolver sistema.
4. Comprobar si el error medio es inferior al 3 % o si la diferencia entre la iteración anterior y la actual es inferior a un hercio. Si se cumple alguna de las dos condiciones anteriores sale del bucle iterativo, sino continua.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

6 ANÁLISIS DE RESULTADOS

El objeto de este apartado es simular una serie de ensayos y analizar la coherencia de los resultados obtenidos. Probaremos las distintas opciones que nos ofrece el programa, explicando lo que hemos ido haciendo y mostrando las salidas gráficas para analizar lo que ha ido ocurriendo.

Comenzaremos probando todas las posibilidades del programa con un modelo de pala de 29 metros de longitud. Una vez vistas las posibilidades del programa para este modelo, se hará una simulación para otro distinto, dejando patente que no sólo vale para un espécimen concreto.

6.1 ESPECIMEN DE 29 METROS DE LONGITUD ENSAYADO EN FLAP

6.1.2 RESONANCIA

Según la teoría de vibraciones, si se excita una viga (en nuestro caso una pala) según su primera frecuencia de vibración, ésta entra en resonancia, fenómeno según el cual las vibraciones se amplifican infinitamente en el caso de que no haya amortiguación. En este caso no serán infinitas porque tenemos al aire haciendo las veces de amortiguador.

Una manera de que la pala entre en resonancia es ejercer sobre un punto de la pala una fuerza oscilatoria cuya frecuencia sea la frecuencia natural de la pala.

Para simular este fenómeno hemos introducido los siguientes datos:

- Masa muerta (365 kg) en el radio 9 m.
- Masa muerta (11 kg) en el radio 28 m.
- Actuador (965kg) con una masa oscilatoria de 400 kg en el radio 14.7 m.
- Paso temporal de 0.04 segundos.
- 50 periodos.
- 100% de la frecuencia natural

Con la opción resolver simulamos el ensayo con la frecuencia natural. Para ello seleccionamos en la casilla de porcentaje de frecuencia natural el 100%.

En la figura 6 (a) aparece el movimiento de la punta de la pala en el caso de resonancia.

Podemos ver como al introducir una fuerza oscilatoria que tenga como frecuencia la frecuencia natural de la pala, la oscilación obtenida es muy extrema. Vemos que en poco tiempo ha alcanzado unos desplazamientos en punta de veinte, lo cual no es factible puesto que la pala mide veintinueve metros.

En la figura 6 (b) aparecen las cargas resultantes en resonancia, que estarían triplicando la resistencia estática de la pala. Es decir, que la pala rompería antes de que se observara lo representado. Por lo tanto, los resultados obtenidos al excitar la pala con una fuerza oscilatoria de frecuencia la frecuencia natural de la pala coinciden con lo esperado.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

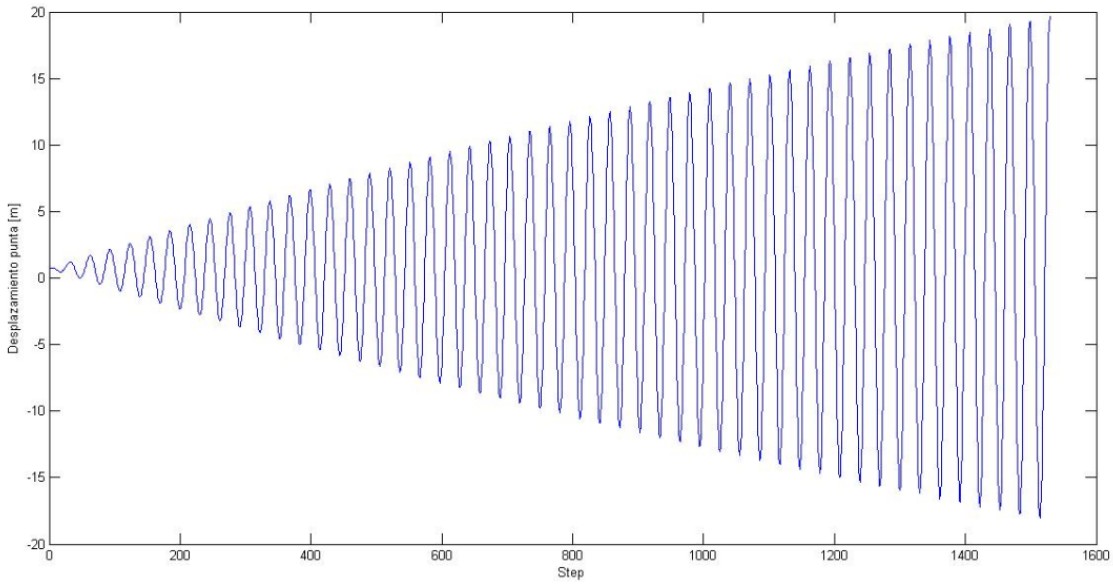


Fig. 6 (a) : Desplazamiento punta de la pala en resonancia.

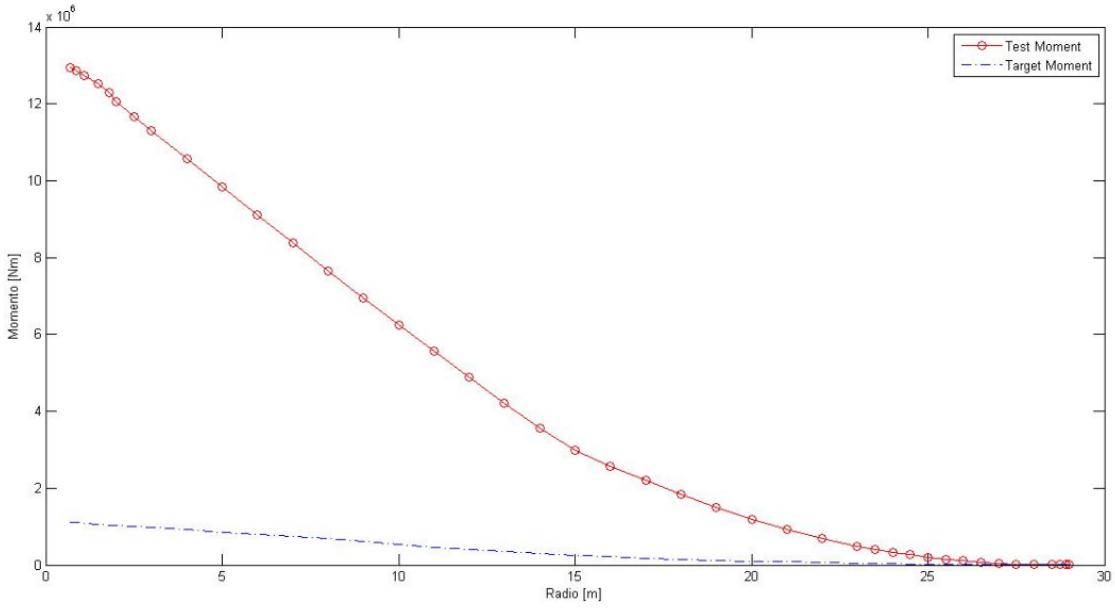


Fig. 6 (b) : Distribución de momentos en resonancia.

6.1.3 RESOLVER

Para analizar la opción resolver vamos a utilizar las mismas masas muertas y actuador que en el caso anterior. Estas no han sido elegidas arbitrariamente, sino que se han seleccionado iguales a las utilizadas en un ensayo real de este modelo unos años atrás. De esta forma hemos podido comprobar que los resultados obtenidos con el programa se ajustan a la realidad.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Para este caso consideramos los siguientes parámetros:

- Paso temporal igual a 0.04 segundos.
- Frecuencia al 96.5 % de la frecuencia natural (5.67 rad/s)
- 75 periodos.

En esta simulación se mostrarán las gráficas de la oscilación de la punta de pala a lo largo del tiempo, la comparativa de momentos y una gráfica en la que se vea la fuerza osciladora introducida en uno de los nodos (Fig. 6 (e)).

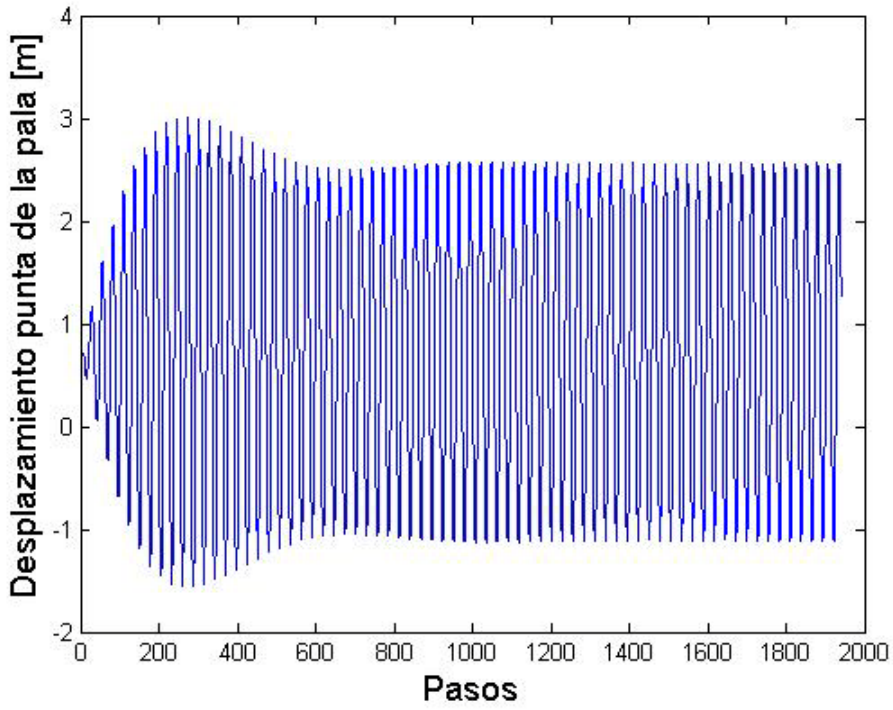


Fig. 6 (c) : Oscilación punta de la pala.

La figura (c) muestra los desplazamientos de la pala a lo largo del tiempo. En un comienzo está oscila de manera transitoria con una forma un tanto peculiar, pero enseguida alcanza el estacionario, vibrando según la frecuencia del actuador y con una amplitud dada.

En la figura 5 (d) comprobamos que con esta frecuencia, estas masas muertas y con el actuador seleccionado, los momentos de ensayo se ajustan a los objetivos. Por lo tanto, hemos simulado un ensayo que se ajusta a la realidad, ya que los resultados coinciden con datos empíricos. Si bien vemos que podríamos elevar ligeramente la frecuencia para ajustarse más a los momentos objetivos.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

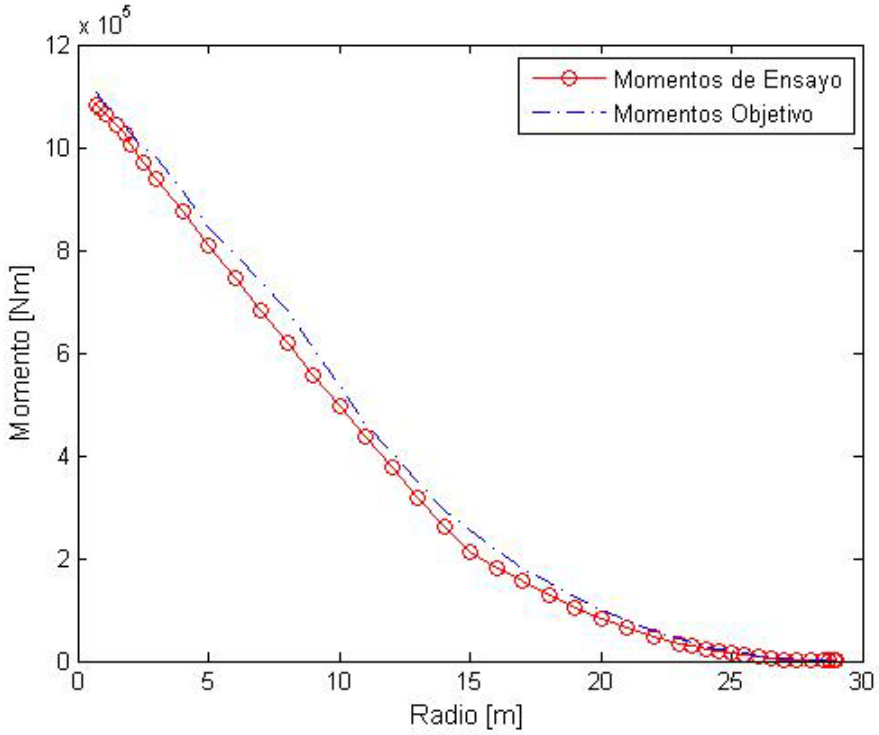


Fig. 6 (d) : Comparativa de momentos.

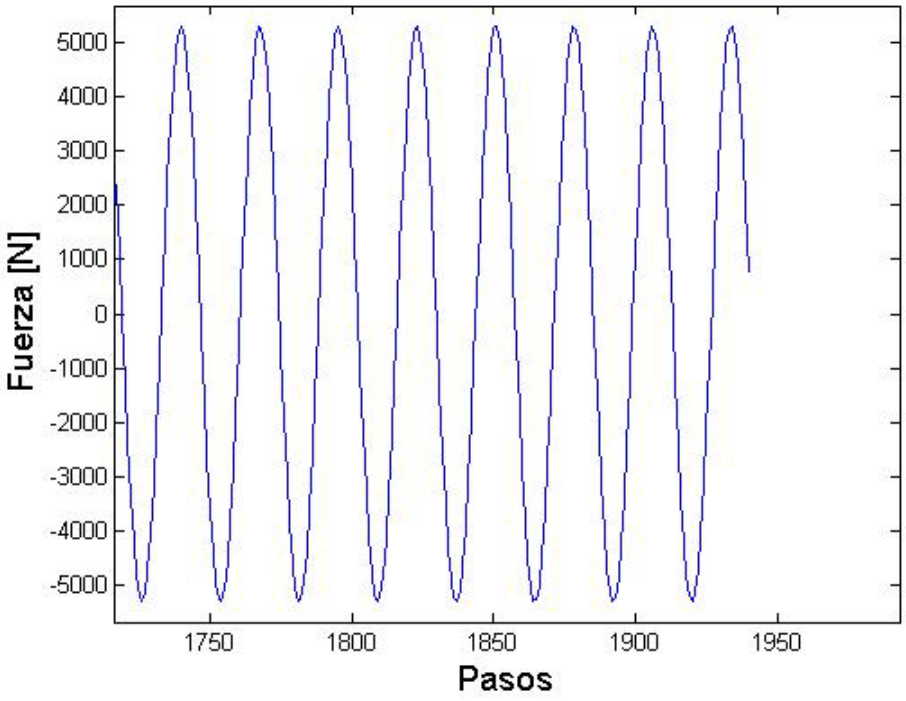


Fig. 6 (e) : Fuerza oscilatoria en uno de los nodos.

6.1.4 OPTIMIZACIÓN

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Una de las opciones que ofrece el programa es la de optimizar la frecuencia de ensayo, de manera que la distribución de momentos obtenida en el mismo se ajuste al máximo a la distribución de momentos objetivo introducida por el usuario en el paso final.

En este caso hemos introducido las mismas masas muertas y el mismo actuador que en los ejemplos anteriores. Además los hemos colocado todos en las mismas posiciones. Por lo tanto, la frecuencia natural resultante ha sido igualmente 0.935 Hz o 5.874 rad/s.

Como paso temporal hemos seleccionado 0.04 segundos, habiendo comprobado previamente que este paso es suficientemente fino para que el resultado que se obtenga sea representativo.

Para ver la evolución de cómo ha ido optimizando la frecuencia de ensayo, se van a ir mostrando en cada paso las gráficas de momentos objetivo y de ensayo. A su vez se irá indicando cual era la frecuencia en cada paso. Además de eso se mostrarán el movimiento de la punta de la pala a lo largo de la simulación para darnos una idea de cómo vibra la pala.

En este caso hemos pedido que la simulación represente 75 periodos de oscilación, tiempo suficiente para que la pala alcance el régimen de vibración estacionario en que estamos interesados.

Primera iteración:

- Frecuencia de ensayo: 93 % de la frecuencia natural (5.464 rad/s).
- Error: -45% (distribución de momentos de ensayo por debajo de la de momentos objetivo.

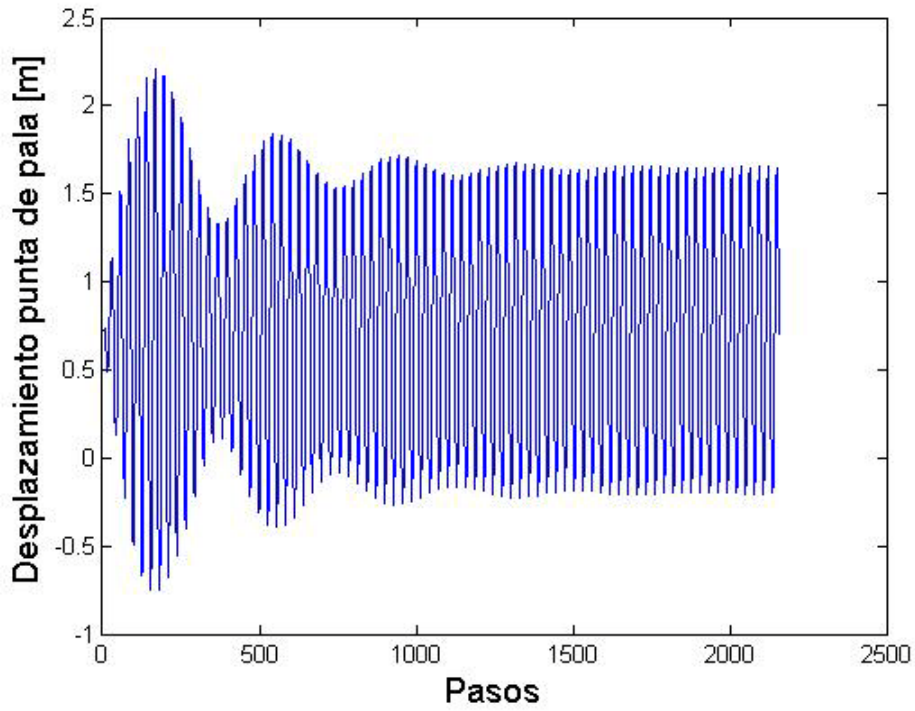


Fig. 6 (f) : Primera iteración punta pala.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

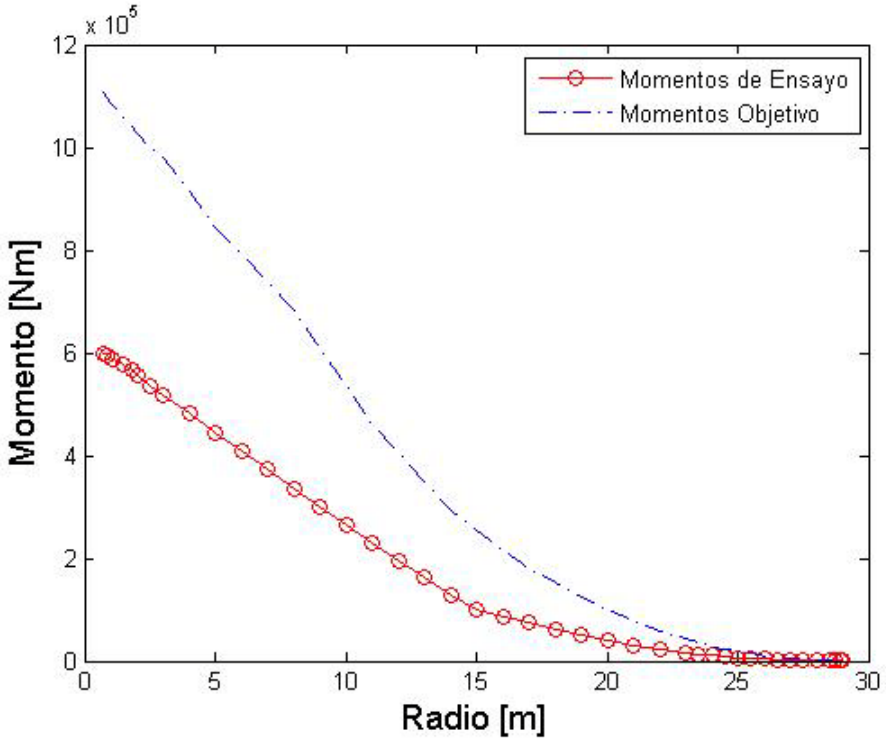


Fig. 6 (g) : Primera iteración comparativa momentos.

En la figura 5 (g) se queda patente que para un 93 % de la frecuencia natural , la distribución de momentos de ensayo está muy por debajo de la distribución objetivo.

El método iterativo para la optimización lo que hace ahora es aumentar la frecuencia para que la distribución de momentos se haga mayor. El programa tiene implementada una cota superior para la frecuencia de ensayo que es el 98.5 % de la frecuencia natural y una inferior que es el 80 % de la misma. En este caso al tener que aumentar la carga toma como cota inferior la frecuencia con la que ha simulado la iteración anterior y calcula el punto medio entre ésta y la cota superior. El resultado es que obtiene una frecuencia para la siguiente iteración de 5.540 rad/s.

Segunda iteración:

- Frecuencia de ensayo: 5.625 rad/s.
- Error: -15%.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

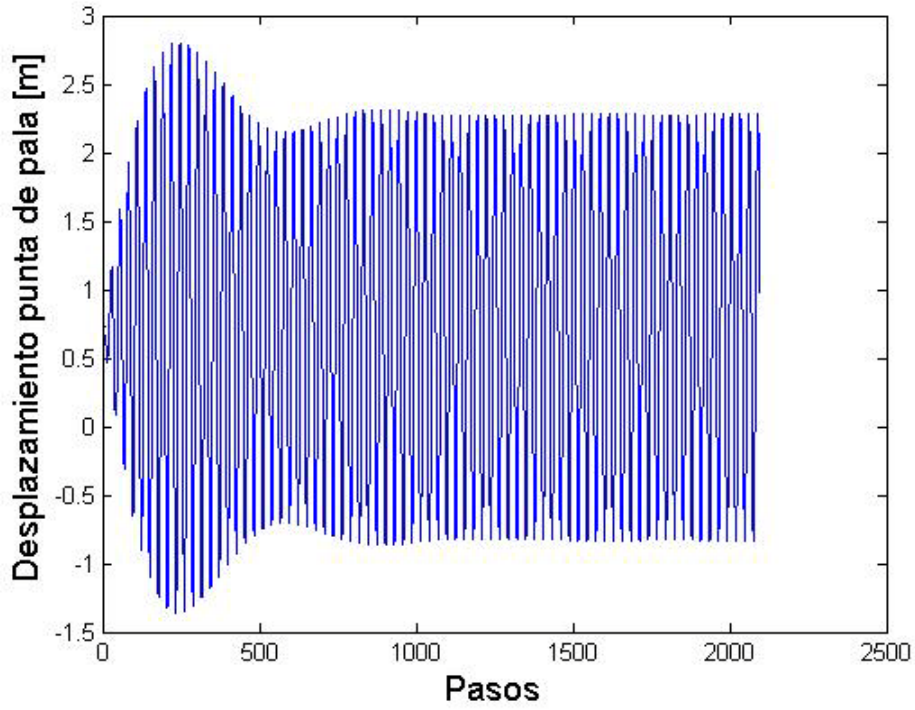


Fig. 6 (h) : Segunda iteración, vibración punta de pala.

Nuevamente la distribución de momentos de ensayo está por debajo de la de momentos objetivo. El error medio de la segunda iteración es del 15 %. En consecuencia el programa toma como cota inferior la frecuencia de la segunda iteración. A continuación calcula el punto medio entre la cota inferior y la cota superior, tomando este como frecuencia para la tercera iteración. Esta frecuencia para la tercera iteración es 5.662 rad/s.

Tercera iteración:

- Frecuencia de ensayo: 5.706 rad/s.
- Error relativo: 11%.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

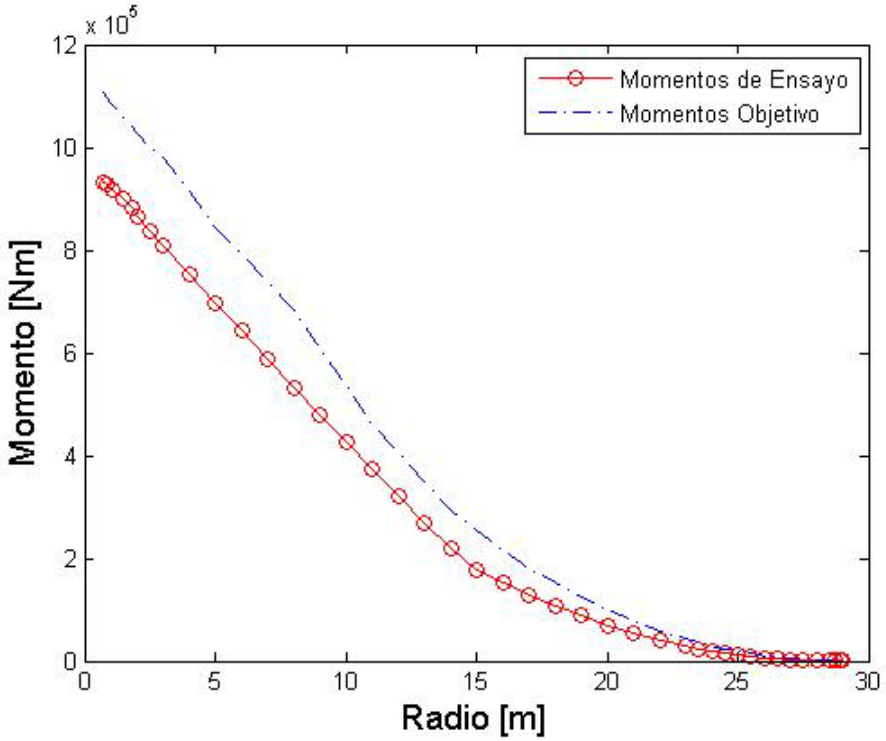


Fig 6 (i): Segunda iteración comparativa de momentos.

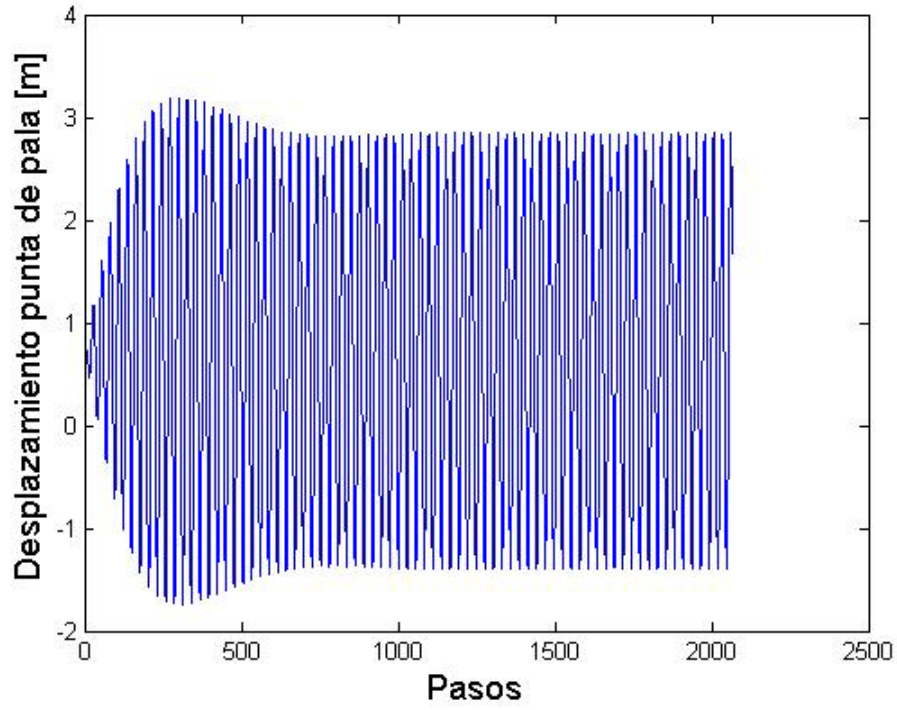


Fig. 6 (j): Vibraciones punta de pala, tercera iteración.

En la tercera simulación la distribución de momentos de ensayo está por encima de la distribución de momentos objetivo (ver Fig. 6 (j)), con lo que para la siguiente iteración habrá que reducir la frecuencia de ensayo. La frecuencia de la tercera iteración se toma como cota máxima, hallándose la

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

de la cuarta iteración como el punto medio entre la cota máxima y la cota mínima. Así, la frecuencia obtenida para la cuarta iteración es 5.665 rad/s.

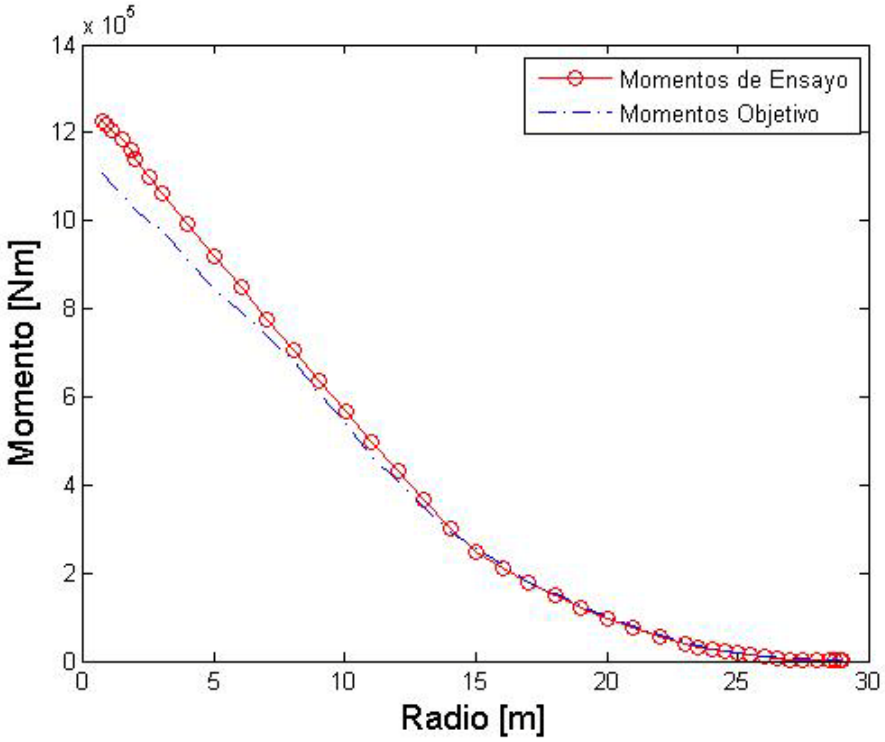


Fig. 5 (k) : Comparativa de momentos, tercera iteración.

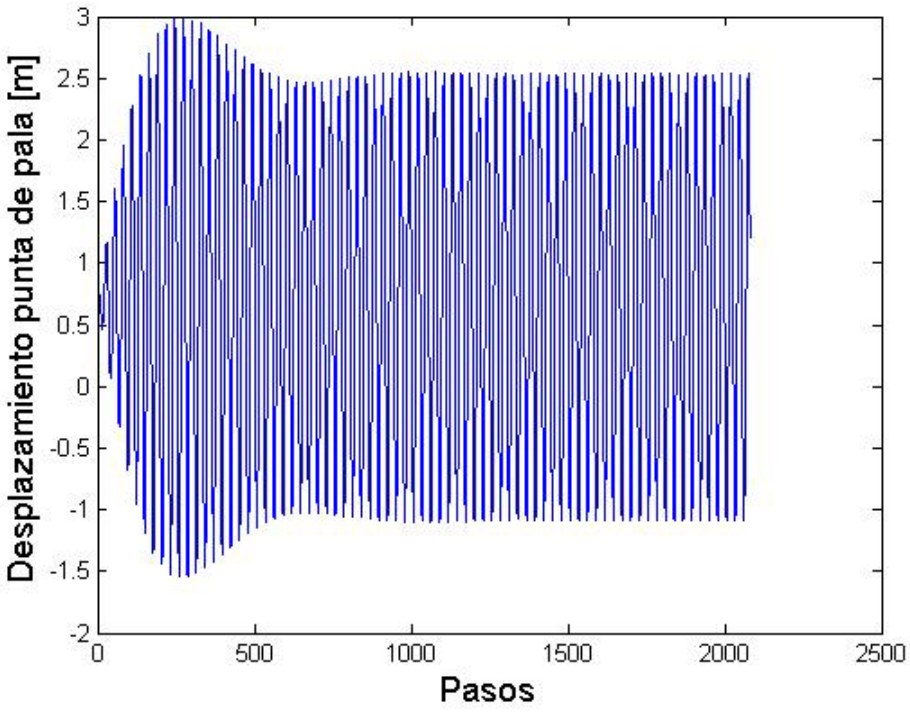


Fig. 6 (l) : Vibraciones punta de pala, cuarta iteración.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Cuarta iteración:

- Frecuencia de ensayo: 5.665 rad/s.
- Error: -2.8 %.

En la simulación de la cuarta iteración, se obtiene un error inferior al 3 %, por lo que al ser éste uno de los criterios de parada se detienen el proceso iterativo dando como frecuencia óptima 5.665 rad/s.

Visto esto podemos concluir que el programa cumple los objetivos para los que fue diseñado:

- Obtiene resultados que se ajustan a la realidad.
- Es capaz de ajustarse a una distribución de momentos objetivo dada optimizando la frecuencia con que debemos excitar la pala.

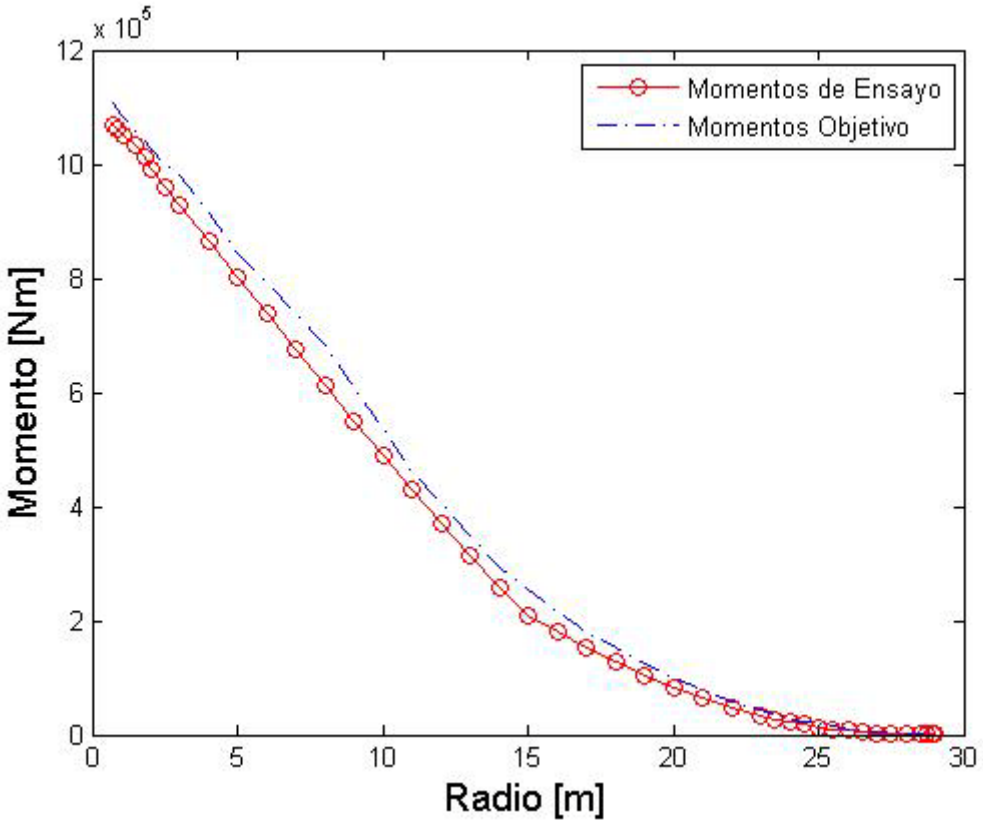


Fig. 6 (m) : Comparativa de momentos, cuarta iteración.

6.2 ESPECÍMEN 44 M DE LONGITUD ENSAYADO EN EDGE

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

Para verificar que la aplicación desarrollada vale para diferentes modelos de pala en este aparrtado vamos a probarlo con una pala de cuarenta y cuatro metros de longitud. Dado que ya se han mostrado las diferentes funciones del programa, en este caso solo se probará la opción resolver. De paso

En este caso no se disponía de una distribución de masas y actuadores conocida de otro ensayo, por lo que se han colocado aleatoriamente. Viendo la figura 6 (o), se observa que se puede mejorar mucho la distribución escogida, así como la frecuencia de ensayo. Al ser el objetivo de esta prueba simplemente ver que el programa funciona para distintos modelos no importa que la distribución de momentos de ensayo quede muy alejada de la de momentos objetivo.

Los actuadores y masas muertas introducidos han sido:

- Actuador en radio 20 metros de 950 kg (350 kg de masa oscilatoria).
- Actuador en radio 34 metros de 750 kg (200 kg de masa oscilatoria).
- Masa muerta en radio 28 metros de 300 kg.
- Masa muerta en radio 39 metros de 50 kg.
- Masa muerta en radio 28 metros de 20 kg.

Los parámetros escogidos para la opción resolver son:

- Paso 0.04 segundos.
- 70 periodos.
- 96.5 % de la frecuencia natural

Los resultados obtenidos se muestran a continuación en las figuras 6(n) y 6(o). De ellas se extrae que habría que disminuir la frecuencia y probablemente cambiar la distribución de las masas muertas y actuadores.

También se observa que le cuesta amortiguar más las frecuencias secundarias, dado que al ser un caso de edge el amortiguamiento aerodinámico es mucho menor.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

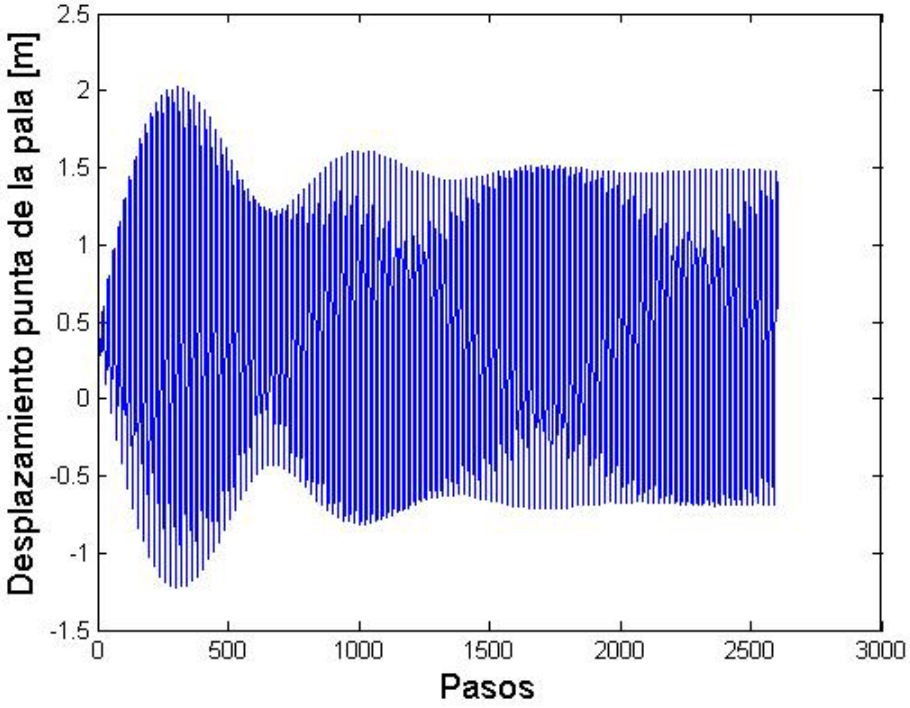


Fig. 6 (n) : Punta de pala de 44 metros

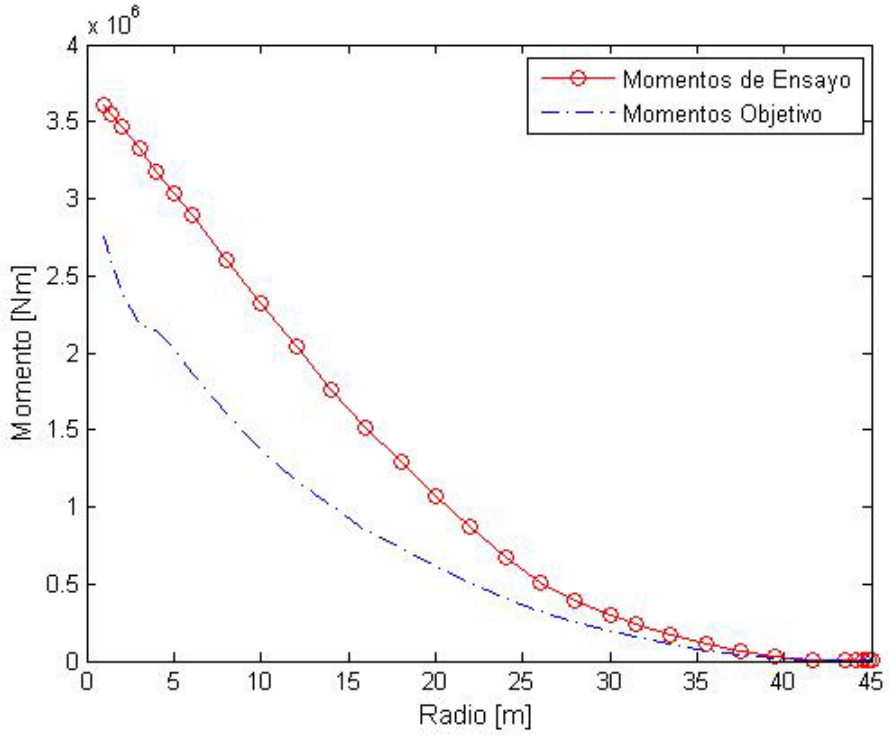


Fig. 6 (o) : Comparativa de momentos espécimen 44 metros

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

7 CONCLUSIONES

El objetivo del proyecto era programar una herramienta que simulase un ensayo a fatiga, obteniéndose los momentos resultantes en el ensayo, objetivo que ha sido alcanzado como se ha expuesto en el apartado 6. Allí vemos como los resultados de nuestra simulación son coherentes con los datos de un ejemplo real.

Además el programa debía ser capaz de optimizar la frecuencia de ensayo a partir de una frecuencia inicial, unas masas muertas y unos actuadores dados, requerimiento que nuevamente queda comprobado en el apartado 6.4.

Otro objetivo no explícito del proyecto era que la herramienta fuese ágil, es decir que se pudieran hacer varias simulaciones en poco tiempo. Este objetivo también se puede decir que se ha logrado, puesto que salvo que quieras realizar una simulación de un tiempo excesivo, el tiempo computacional es razonable. No tiene sentido realizar una simulación excesivamente larga puesto que la pala alcanza el estacionario rápidamente y a partir de ahí se repite continuamente el movimiento.

En otro orden de cosas, actualmente los ensayos de fatiga se realizan en una sola dirección al mismo tiempo, como se contempla en nuestro proyecto. Además, un ensayo a fatiga puede durar un mes e incluso dos, pudiéndose alargar la duración en un futuro debido a la tendencia a desarrollar palas mayores (tienen frecuencias naturales menores).

En consecuencia, se están empezando a desarrollar ensayos en los que se carga la pala en dos direcciones simultáneamente, reduciendo considerablemente los tiempos de ensayo y, por ende, los costes. Este modelo de ensayo no se ha contemplado porque todavía está en fase experimental. A pesar de ello, parece claro que en un futuro cercano se acabará imponiendo este tipo de ensayo, con lo que otra mejora sería incorporar esa aplicación al programa.

Con todo, consideramos que se han cumplido satisfactoriamente los objetivos para los que se planteó el proyecto, si bien hemos encontrado algunas mejoras que podrían implementarse para complementar la aplicación desarrollada.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

8 BIBLIOGRAFÍA

[1] O.C Zienkiewicz & R.L Taylor, *The Finite Element Method*, Butterworth-Heinemann, 2005.

[2] Y.K.Cheung & A.Y.T.Leung, *Finite Element Methods in Dynamics*, Kluwer Academic Publishers, 1991.

[3] K.Kythe, *An Introduction to Linear and Non Linear Finiete Element Analysis*, Birkhäuser, 2003.

[4] Javier Sayas, *Advanced Finite Element* (apuntes en español), 2005.

[5] Javier Sayas, *Mathematical Models in Mechanics* (apuntes en español), 2006.

[6] J.He & Z.Fu, *Modal Analysis*, Butterworth-Heinemann, 2001.

[7] L.Meirovitch, *Fundamentals of Vibrations*, McGraw-Hill, 2001.

[8] S.G.Kelly, *Mechanical Vibrations*, McGraw-Hill, 1996.

[9] J.Shigley, C.Mischke & R.Buydnas, *Mechanical Engineering Design*, McGraw-Hill, 2004.

[10] D. Schütz & J.J. Gerharz, *Fatigue Strength of a Fibre-Reinforced Material*, Composites, 1977.

[11] D. Schütz, '*Standardized Stress-Time Histories - An Overview', Development of Fatigue Loading Spectra*, American Society for Testing and Materials, Philadelphia, 1989.

[12] L.O.Berrocal, *Resistencia de Materiales*, McGraw-Hill, 2007.

[13] R.P.L Nijsssen, *Fatigue Life Prediction and Strength Degradation of Wind Turbine Rotor Blade Composites*, KC-WMC, 2006.

[14] T.Burton, D.Sharpe, N.Jenkins y E.Bossanyi, *Wind Energy Handbook*, John Whiley&Sons, 2001.

[15] IEC TS 61400-23, *Full-scale structural testing of rotor blades*, 2002.

[16] Germanischer Lloyd WindEnergie GmbH, *Guideline for the Certification of Wind Turbines*, 2003.

[17] Det Norske Veritas, *Guidelines for Design of Wind Turbines*, 2002.

[18] P.Brøndsted, H.Lilholt,, A.Lystrup., *Composite Materials for Wind Power Turbine Blades*, Annu. Rev. Mater. Res., Vol. 35, 2005, pp. 505-538

[19] N.K.Wahl, *Spectrum fatigue lifetime and residual strength for fiberglass laminates*, Ph.D. Thesis, Montana State University, Bozeman, 2001.

[20] DS472, *Loads and Safety of Wind Turbine Construction*, Danish Standards,2006.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

[21] I.Alonso Mayo, B.Cano y M.J.Morera, *Stable Runge-Kutta-Nyström for dissipative stiff problems*, Numer. Algor. 42, pp. 193-203, 2006.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

SIMULACIÓN DE ENSAYOS

A FATIGA PARA PALAS DE

AEROGENERADORES

Manual de usuario

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A. MANUAL DE USUARIO

A.1 INTRODUCCIÓN

Este manual describe el funcionamiento general de la herramienta programada en Matlab “Simulación de ensayos de fatiga”. A lo largo del mismo se detalla los pasos a seguir para simular un ensayo a fatiga y calcular la distribución de momentos de rango aplicada sobre la pala a lo largo del ensayo.

El documento está estructurado de la siguiente manera:

1. Menú principal.
2. Propiedades de la pala.
3. Actuadores y masas muertas.
4. Análisis Modal.
5. Resolución temporal.
6. Resolución estacionario.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A.2 MENU PRINCIPAL

Una vez se ha ejecutado la aplicación, el menú principal se presenta en pantalla. Éste presenta el siguiente aspecto:

Fig. A (a) : Menú principal.

En la parte izquierda de la pantalla se encuentran los botones referentes a los distintos bloques del programa. El orden en que aparecen los botones no es aleatorio y es indispensable ejecutarlos de arriba hacia abajo. Esto se debe a que por ejemplo las propiedades de la pala, introducidas al pulsar el botón “Propiedades de la pala”, son necesarias en los pasos subsiguientes y así sucesivamente con el resto de botones.

Solo en el caso de los últimos dos botones se puede pulsar indistintamente uno u otro en función de que se quiera realizar una resolución temporal o que por el contrario se busque simplemente una resolución del sistema estacionario.

Lo que ocurre al pulsar cada uno de los botones se explica en los apartados siguientes correspondientes.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A.3 PROPIEDADES DE LA PALA

Al hacer *click* sobre el botón "Propiedades de pala", se nos pide seleccionar el fichero con las propiedades de la pala que se desea ensayar. Un ejemplo de esto aparece en la figura A (b).

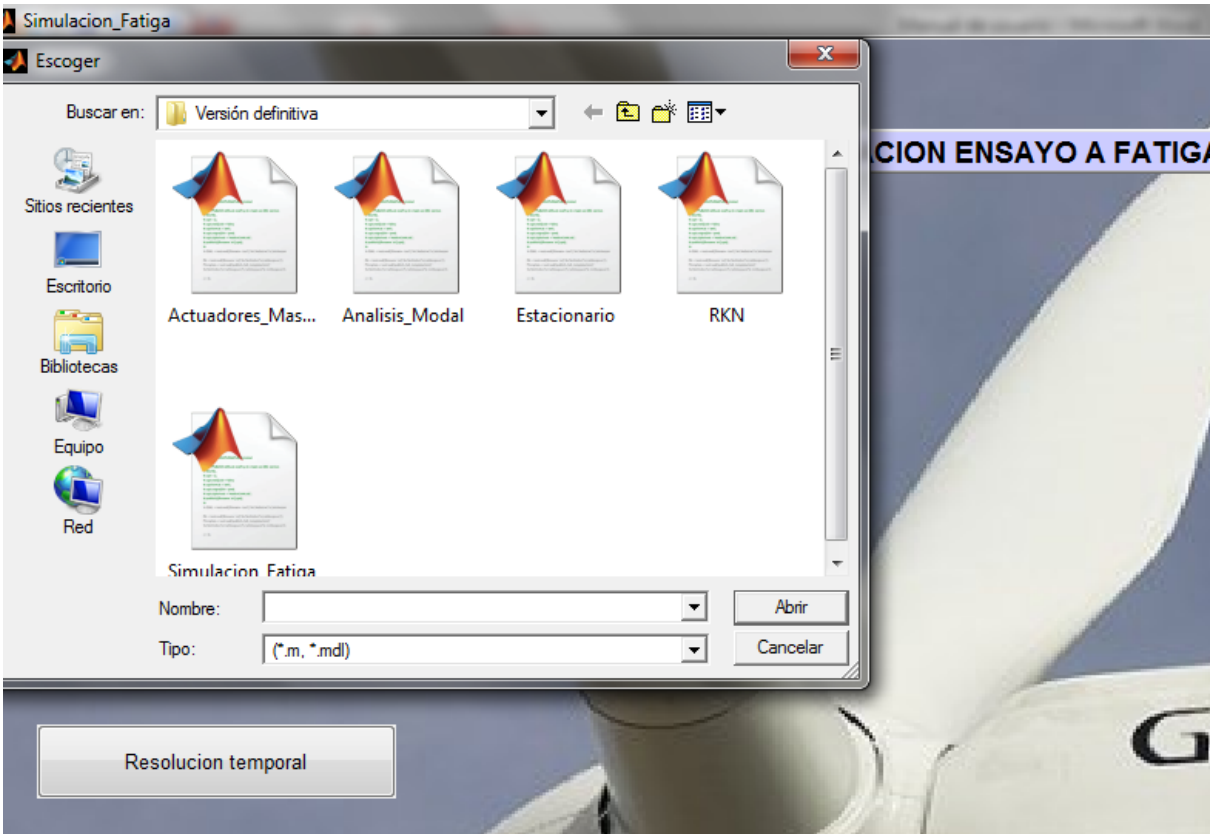


Fig. A (b) : Cargar fichero con propiedades de la pala.

El fichero seleccionado deberá tener extensión xls o xlsx. En el deberán aparecer las propiedades de la pala para cada una de las secciones de las que disponemos de propiedades, ordenadas de la siguiente manera:

- 1ª columna: radios en los que disponemos de propiedades.
- 2ª columna: masa lineal para cada uno de los radios de la primera columna.
- 3ª columna: rigidez en la dirección de flap.
- 4ª columna: rigidez en la dirección de edge.
- 5ª columna: cuerda de la pala.
- 6ª columna: espesor de la pala.

Las primeras filas en las que aparezcan caracteres no serán tenidas en cuenta. Un ejemplo del fichero a cargar en este apartado aparece en la figura A (4). En esta podemos ver que en las dos primeras líneas aparecen los títulos de las columnas, así como sus unidades. Estas dos filas al ser caracteres no son tenidas en cuenta por el programa y por tanto no son necesarias.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

	A	B	C	D	E	F	G
1	Radius [m]	Lindensity	Dflap	Dedge	Cuerda	Espesor	
2	[m]	[kg/m]	[Nm2]	[Nm2]	[m]	[m]	
3	0,7	561,6	1166532000	1166551000	2,5	2,5	
4	0,85	697,78	1187147000	1186430000	2,6	2,6	
5	1,1	505,88	963077000	962316000	2,7	2,68	
6	1,5	255,85	587336000	608003000	3,1	3	
7	1,8	166,52	345420000	341173000	3,2	2,9	
8	2	143,64	279901000	287220000	3,1	2,8	
9	2,5	139,24	256114000	282120000	2,9	1,8	
10	3	147,65	245650000	292508000	2,85	1,3	
11	4	148,17	235415000	283637000	2,8	0,9	
12	5	141,78	208458000	281876000	2,8	0,52	
13	6	129,56	158854000	269520000	2,7	0,53	

Fig. A (c) : Fichero estándar con propiedades de la pala.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A.4 ACTUADORES Y MASAS MUERTAS

Una vez han sido introducidas las propiedades de la pala a ensayar, lo siguiente, es seleccionar los actuadores y las masas muertas con las que se quiere simular el ensayo.

Al hacer *click* sobre el botón "Actuadores y masas muertas" aparecen en la pantalla dos tablas a rellenar por el usuario, tal como se muestra en la figura A (d). En una de las tablas se introduce toda la información referente a los actuadores y en la otra la relacionada con las masas muertas.

Fig. A (d) : Actuadores y masas muertas.

Ambas tablas tienen diez filas, luego se pueden introducir hasta diez actuadores y diez masas muertas. Solo hay que rellenar tantas filas como actuadores queramos en un caso e idem para las masas muertas.

Para los actuadores hay que introducir el radio en el que los queremos colocar, su masa total y su masa oscilatoria. Dentro de la masa total del oscilador debe estar incluida tanto la masa oscilatoria como la masa no oscilatoria del mismo.

En el caso de las masas muertas hay que especificar el radio en el que están colocadas así como su masa.

Al igual que con las propiedades de la pala, se deben introducir todos los datos de los actuadores y de las masas muertas en sistema internacional.

Se debe tener especial precaución en no introducir masas muertas y actuadores en el mismo radio. Notar que esto no tiene sentido físico, porque o hay un actuador o hay una masa muerta.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A.5 ANÁLISIS MODAL

Después de haber introducido las propiedades de la pala, los actuadores y las masas muertas, se procede a calcular la frecuencia natural de la pala (teniendo en cuenta masas muertas y actuadores). Para ello debemos clicar sobre el botón "Análisis modal", apareciendo en pantalla lo mostrado en la figura A (e).

Fig. A (e) : Análisis modal.

En esta pantalla se debe escoger el número de elementos con que se quiere mallar la pala. A mayor número de elementos mejor será la solución, pero si éste es excesivo podría ralentizar el programa. Se recomienda escoger un número de elementos $n \geq \frac{L}{200}$, donde L es la longitud de la pala en milímetros.

Además es en este apartado donde el usuario debe escoger si quiere simular un ensayo a fatiga en la dirección de *flap* o en la de *edge*, para lo cual dispone de un panel el que debe seleccionar uno u otro ensayo.

Cuando se tiene seleccionado el número de elementos y el tipo de ensayo a simular seleccionados lo siguiente es clicar sobre el botón "Frecuencia de ensayo", tras lo cual el programa nos muestra en esta misma pantalla la frecuencia natural de la pala, tanto en radianes como en hercios.

El usuario puede volver calcular la frecuencia natural tantas veces como quiera modificando el número de elementos o el tipo de ensayo sin salirse de esta pantalla. Una vez esté satisfecho con el resultado obtenido podrá volver al menú principal clicando sobre el botón aceptar, quedándose el número de elementos, la frecuencia y el tipo de ensayo del último cálculo grabados.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A.6 RESOLUCIÓN TEMPORAL

Si seleccionamos la opción “Resolución temporal”, pasaremos a resolver el problema en el tiempo. La pantalla que aparece al clicar esa opción se muestra en la figura A (f). Como solución alternativa a la resolución temporal podemos seleccionar “Resolución estacionaria” explicada en el punto A.7, que no tiene en cuenta ni el transitorio ni las pérdidas aerodinámicas.

Fig. A (f) : Resolución temporal.

Una vez estamos en la pantalla de la figura A (f), lo primero que debe hacerse es clicar el botón de cargas objetivo y cargar un archivo xls o xlsx que contenga las cargas que se quieren alcanzar en el ensayo. Un ejemplo de este fichero aparece en la figura A (g).

	A	B	C	D
1	Radio	Mrango	Mmedio	
2	[m]	[Nm]	[Nm]	
3	0,7	1106440	402000	
4	0,85	1097320	397000	
5	1,1	1082120	390000	
6	1,5	1057800	365000	
7	1,8	1039560	341000	
8	2	1027400	312000	

Fig. A (g) : Cargas objetivo.

Nuevamente el orden de las columnas debe ser el mostrado en la figura A (g).

En esta pantalla se debe elegir a la frecuencia a la que queremos realizar el ensayo. Esta selección la hacemos por porcentaje de la frecuencia natural en la casilla correspondiente. Es decir, si la frecuencia natural de la pala (con actuadores y masas muertas) es de un hercio y yo quiero simular mi ensayo a 0.95 hercios, pondré en la casilla correspondiente 95.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

También será necesario seleccionar el paso temporal, p, para resolver el problema en el tiempo. Cuanto menor sea p, mejor será la solución. Sin embargo un p muy pequeño podría ralentizar mucho la resolución.

Por último antes de la resolución, debemos escoger el número de periodos de oscilación que queremos representar. Se debería escoger un número de periodos suficientes para alcanzar el estacionario, debido a que este es el estado que nos interesa y no el transitorio inicial.

Cuando se ha elegido todo lo anterior podemos resolver el sistema de dos maneras:

- “Resolver”.
- “Optimizar”.

Si se escoge la opción resolver el programa nos simula el ensayo para la frecuencia escogida y nos da como salida una comparativa de los momentos de rango obtenidos en la simulación frente a los objetivo.

Si por el contrario elegimos la opción optimizar el programa optimiza la frecuencia de ensayo mediante un proceso iterativo para aproximar al máximo la curva de momentos de ensayo a la distribución de momentos objetivo. Este proceso iterativo no para hasta que las dos distribuciones sean lo suficientemente parecidas (error < 1%), o que la diferencia de las frecuencias obtenidas de dos iteraciones sucesivas inferior a una tolerancia establecida (< 0.01*ω).

Una vez elijamos una opción u otra el programa resolverá el problema y volverá al menú principal. Una vez resuelto proporciona las siguientes salidas:

- Fichero Excel con las distribuciones medias y de rango, tanto las obtenidas en la simulación como las objetivo. Además también mostrará la frecuencia de ensayo.
- Gráfica comparativa de momentos de ensayo frente a momentos objetivo.
- Gráfica de desplazamientos de la punta de la pala durante la simulación.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

A.7 RESOLUCIÓN ESTACIONARIA

La alternativa a realizar una resolución temporal es la de resolver el sistema en el estacionario. Tiene el inconveniente de que no se contemplan las pérdidas aerodinámicas. Para elegir esta opción se debe clicar el botón “Resolución estacionaria”, apareciendo la pantalla que se muestra en la figura A (h).

Fig. A (h) : Resolución del estacionaria.

Elegida esta opción habrá que cargar los momentos objetivo y elegir la frecuencia natural, de la misma manera que se explica para la resolución temporal. La diferencia radica en que en este caso al resolver el estacionario y no hacer una resolución en el tiempo, no se introduce un paso temporal.

Las opciones resolver y optimizar son análogas a las de la “Resolución temporal”.

Así mismo las salidas que ofrece esta opción son las mismas que para la “Resolución temporal”, con la diferencia de que en este caso en vez de dibujar los desplazamientos de la punta de la pala a lo largo del tiempo de simulación, lo que grafica son treinta segundos de la evolución del estacionario para la punta de la pala.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

B CÓDIGO FUENTE

B.1 SIMULACIÓN FATIGA

```
function varargout = Simulacion_Fatiga(varargin)

% SIMULACION_FATIGA M-file for Simulacion_Fatiga.fig

%      SIMULACION_FATIGA, by itself, creates a new SIMULACION_FATIGA or
raises the existing

%      singleton*.

%

%      H = SIMULACION_FATIGA returns the handle to a new SIMULACION_FATIGA
or the handle to

%      the existing singleton*.

%

%      SIMULACION_FATIGA('CALLBACK',hObject,eventData,handles,...) calls
the local

%      function named CALLBACK in SIMULACION_FATIGA.M with the given input
arguments.

%

%      SIMULACION_FATIGA('Property','Value',...) creates a new
SIMULACION_FATIGA or raises the

%      existing singleton*.  Starting from the left, property value pairs
are

%      applied to the GUI before Simulacion_Fatiga_OpeningFcn gets called.
An

%      unrecognized property name or invalid value makes property
application

%      stop.  All inputs are passed to Simulacion_Fatiga_OpeningFcn via
varargin.

%

%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one

%      instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
% Edit the above text to modify the response to help Simulacion_Fatiga


% Last Modified by GUIDE v2.5 20-Nov-2010 16:53:10


% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Simulacion_Fatiga_OpeningFcn, ...
                   'gui_OutputFcn',  @Simulacion_Fatiga_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end


if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT




% --- Executes just before Simulacion_Fatiga is made visible.
function Simulacion_Fatiga_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Simulacion_Fatiga (see VARARGIN)
a=imread('fatiguefig1.jpg');
image(a)
axis off
% Choose default command line output for Simulacion_Fatiga
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);




% UIWAIT makes Simulacion_Fatiga wait for user response (see UIRESUME)
% uiwait(handles.figure1);



% --- Outputs from this function are returned to the command line.
function varargout = Simulacion_Fatiga_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in Blade_Properties.
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
function Blade_Properties_Callback(hObject, eventdata, handles)
% hObject    handle to Blade_Properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Lee las propiedades del archivo xls ordenadas en cuatro columnas(Radio,
% masa por unidad de longitud,rigidez en flap, rigidez en edge, cuerda
% y espesor)
[FileName Path]=uigetfile({'*.m;*.mdl'}, 'Escoger');
global Masa_Lineal Radio Rigidez_Flap Rigidez_Edge Cuerda_Pala Espesor_Pala
A=xlsread(FileName);
Radio=A(:,1);
Masa_Lineal=A(:,2);
Rigidez_Flap=A(:,3);
Rigidez_Edge=A(:,4);
Cuerda_Pala=A(:,5);
Espesor_Pala=A(:,6);
% --- Executes on button press in Actuators_and_dead_loads.
function Actuators_and_dead_loads_Callback(hObject, eventdata, handles)
% hObject    handle to Actuators_and_dead_loads (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Actuadores_Masas_Muertas



% --- Executes on button press in Modal_Analysis.
function Modal_Analysis_Callback(hObject, eventdata, handles)
% hObject    handle to Modal_Analysis (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Analisis_Modal
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
% --- Executes on button press in RKN.
function RKN_Callback(hObject, eventdata, handles)
% hObject    handle to RKN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
RKN



% --- Executes on button press in Stationary.
function Stationary_Callback(hObject, eventdata, handles)
% hObject    handle to Stationary (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Estacionario
```

B.2 ACTUADORES Y MASAS MUERTAS

```
function varargout = Actuadores_Masas_Muertas(varargin)
% ACTUADORES_MASAS_MUERTAS M-file for Actuadores_Masas_Muertas.fig
%      ACTUADORES_MASAS_MUERTAS, by itself, creates a new
ACTUADORES_MASAS_MUERTAS or raises the existing
%      singleton*.
%
%      H = ACTUADORES_MASAS_MUERTAS returns the handle to a new
ACTUADORES_MASAS_MUERTAS or the handle to
%      the existing singleton*.
%
%      ACTUADORES_MASAS_MUERTAS('CALLBACK',hObject,eventData,handles,...)
calls the local
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
%      function named CALLBACK in ACTUADORES_MASAS_MUERTAS.M with the given
input arguments.

%

%      ACTUADORES_MASAS_MUERTAS('Property','Value',...) creates a new
ACTUADORES_MASAS_MUERTAS or raises the

%      existing singleton*.  Starting from the left, property value pairs
are

%      applied to the GUI before Actuadores_Masas_Muertas_OpeningFcn gets
called.  An

%      unrecognized property name or invalid value makes property
application

%      stop.  All inputs are passed to Actuadores_Masas_Muertas_OpeningFcn
via varargin.

%

%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one

%      instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES



% Edit the above text to modify the response to help
Actuadores_Masas_Muertas



% Last Modified by GUIDE v2.5 20-Nov-2010 21:17:54



% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;

gui_State = struct('gui_Name',       mfilename, ...

                   'gui_Singleton',  gui_Singleton, ...

                   'gui_OpeningFcn', @Actuadores_Masas_Muertas_OpeningFcn,
...

                   'gui_OutputFcn',  @Actuadores_Masas_Muertas_OutputFcn,
...

                   'gui_LayoutFcn',  [] , ...

                   'gui_Callback',   []);
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end


if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT




% --- Executes just before Actuadores_Masas_Muertas is made visible.
function Actuadores_Masas_Muertas_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Actuadores_Masas_Muertas (see
VARARGIN)
a=imread('fatiguefig1.jpg');
image(a)
axis off
% Choose default command line output for Actuadores_Masas_Muertas
handles.output = hObject;


% Update handles structure
guidata(hObject, handles);
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
% UIWAIT makes Actuadores_Masas_Muertas wait for user response (see
UIRESUME)

% uiwait(handles.figure1);



% --- Outputs from this function are returned to the command line.

function varargout = Actuadores_Masas_Muertas_OutputFcn(hObject, eventdata,
handles)

% varargout  cell array for returning output args (see VARARGOUT);

% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)


% Get default command line output from handles structure

varargout{1} = handles.output;




function edit1_Callback(hObject, eventdata, handles)

% hObject    handle to edit1 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of edit1 as text

%        str2double(get(hObject,'String')) returns contents of edit1 as a
double



% --- Executes during object creation, after setting all properties.

function edit1_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit1 (see GCBO)
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a
double



% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

    set(hObject,'BackgroundColor','white');

end




% --- Executes on button press in pushbutton1.

function pushbutton1_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton1 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

global Espesor Espesor_Pala Cuerda_Pala Rigidez_Flap Rigidez_Edge
Masa_Lineal Radio Radio_Actuador1 Masa_Actuador1 Masa_Osciladora1
Radio_MasaMuerta1 Masa_Muerta1 Radius Linear_Mass Flap_Stiffness
Edge_Stiffness Cuerda Concentrated_Mass Masa_Osciladora Masa_Muerta
Radio_Actuador

%Tomo los valores númericos de las tablas de actuadores y de masas muertas

con=1;

for i=1:10

    if isnan(Radio_Actuador1(i))==0

        Radio_Actuador(con)=Radio_Actuador1(i);

        Masa_Actuador(con)=Masa_Actuador1(i);

        Masa_Osciladora(con)=Masa_Osciladora1(i);

        con=con+1;

    end;

end;

con_1=1;

for i=1:10

    if isnan(Radio_MasaMuerta1(i))==0

        Radio_MasaMuerta(con_1)=Radio_MasaMuerta1(i);

        Masa_Muerta(con_1)=Masa_Muerta1(i);

        con_1=con_1+1;
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
    end;
end;


Tamano=length(Radio);
% Ordeno las masas muertas de menor a mayor radio
for i=1:length(Masa_Muerta)
    Min=Radio_MasaMuerta(i);
    MasaMin=Masa_Muerta(i);
    jmin=i;
    for j=i:length(Masa_Muerta)
        if Radio_MasaMuerta(j)<Min
            Min=Radio_MasaMuerta(j);
            MasaMin=Masa_Muerta(j);
            jmin=j;
        end;
    end;
    Radio_MasaMuerta(jmin)=Radio_MasaMuerta(i);
    Masa_Muerta(jmin)=Masa_Muerta(i);
    Radio_MasaMuerta(i)=Min;
    Masa_Muerta(i)=MasaMin;
end;
% Ordeno los actuadores de menor a mayor radio
for i=1:length(Masa_Actuador)
    Min=Radio_Actuador(i);
    MasaMin=Masa_Actuador(i);
    MasaoscMin=Masa_Osciladora(i);
    jmin=i;
    for j=i:length(Masa_Actuador)
        if Radio_Actuador(j)<Min
            Min=Radio_Actuador(j);
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
            MasaMin=Masa_Actuador(j);
            MasaoscMin=Masa_Osaciladora(j);
            jmin=j;
        end;
    end;
    Radio_Actuador(jmin)=Radio_Actuador(i);
    Masa_Actuador(jmin)=Masa_Actuador(i);
    Masa_Osciladora(jmin)=Masa_Osciladora(i);
    Radio_Actuador(i)=Min;
    Masa_Actuador(i)=MasaMin;
    Masa_Osciladora(i)=MasaoscMin;
end;
% Las propiedades de la pala son interpoladas para las secciones en las que
% hay masas muertas o actuadores, pero no tengo las propiedades definidas
 Radius(1)=Radio(1);
 Linear_Mass(1)=Masa_Lineal(1);
 Flap_Stiffness(1)=Rigidez_Flap(1);
 Edge_Stiffness(1)=Rigidez_Edge(1);
 Cuerda(1)=Cuerda_Pala(1);
 Espesor(1)=Espesor_Pala(1);
 Concentrated_Mass(1)=0;
cont=2;
cont_1=1;
cont_2=1;
Tamano=length(Radio);

for i=2:Tamano
    cepo=1;
    if cont_1<=length(Radio_MasaMuerta)
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
        if (Radio_MasaMuerta(cont_1)<Radio(i)) &&
(Radio_MasaMuerta(cont_1)>Radio(i-1))

            if cont_2<=length(Radio_Actuador)

             if
(Radio_Actuador(cont_2)<Radio_MasaMuerta(cont_1))&&(Radio_Actuador(cont_2)>
Radio(i-1))

                 Radius(cont)=Radio_Actuador(cont_2);

                 Linear_Mass(cont)=Masa_Lineal(i-1)+(Radio_Actuador(cont_2)-
Radio(i-1))*(Masa_Lineal(i)-Masa_Lineal(i-1))/(Radio(i)-Radio(i-1));

                 Flap_Stiffness(cont)=Rigidez_Flap(i-
1)+(Radio_Actuador(cont_2)-Radio(i-1))*(Rigidez_Flap(i)-Rigidez_Flap(i-
1))/(Radio(i)-Radio(i-1));

                 Edge_Stiffness(cont)=Rigidez_Edge(i-
1)+(Radio_Actuador(cont_2)-Radio(i-1))*(Rigidez_Edge(i)-Rigidez_Edge(i-
1))/(Radio(i)-Radio(i-1));

                 Cuerda(cont)=Cuerda_Pala(i-1)+(Radio_Actuador(cont_2)-
Radio(i-1))*(Cuerda_Pala(i)-Cuerda_Pala(i-1))/(Radio(i)-Radio(i-1));

                 Espesor(cont)=Espesor_Pala(i-1)+(Radio_Actuador(cont_2)-
Radio(i-1))*(Espesor_Pala(i)-Espesor_Pala(i-1))/(Radio(i)-Radio(i-1));

                 Concentrated_Mass(cont)=Masa_Actuador(cont_2);

                 cont=cont+1;

                 cont_2=cont_2+1;

             end;

            end;

             Radius(cont)=Radio_MasaMuerta(cont_1);

             Linear_Mass(cont)=Masa_Lineal(i-1)+(Radio_MasaMuerta(cont_1)-
Radio(i-1))*(Masa_Lineal(i)-Masa_Lineal(i-1))/(Radio(i)-Radio(i-1));

             Flap_Stiffness(cont)=Rigidez_Flap(i-1)+(Radio_MasaMuerta(j)-
Radio(i-1))*(Rigidez_Flap(i)-Rigidez_Flap(i-1))/(Radio(i)-Radio(i-1));

             Edge_Stiffness(cont)=Rigidez_Edge(i-1)+(Radio_MasaMuerta(j)-
Radio(i-1))*(Rigidez_Edge(i)-Rigidez_Edge(i-1))/(Radio(i)-Radio(i-1));

             Cuerda(cont)=Cuerda_Pala(i-1)+(Radio_MasaMuerta(cont_1)-
Radio(i-1))*(Cuerda_Pala(i)-Cuerda_Pala(i-1))/(Radio(i)-Radio(i-1));

             Espesor(cont)=Espesor_Pala(i-1)+(Radio_MasaMuerta(cont_1)-
Radio(i-1))*(Espesor_Pala(i)-Espesor_Pala(i-1))/(Radio(i)-Radio(i-1));

             Concentrated_Mass(cont)=Masa_Muerta(cont_1);

             cont=cont+1;

             cont_1=cont_1+1;
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
        elseif cont_2<=length(Radio_Actuador)

             if (Radio_Actuador(cont_2)<Radio(i)) &&
(Radio_Actuador(cont_2)>Radio(i-1))

                 Radius(cont)=Radio_Actuador(cont_2);

                 Linear_Mass(cont)=Masa_Lineal(i-
1)+(Radio_Actuador(cont_2)-Radio(i-1))*(Masa_Lineal(i)-Masa_Lineal(i-
1))/(Radio(i)-Radio(i-1));

                 Flap_Stiffness(cont)=Rigidez_Flap(i-
1)+(Radio_Actuador(cont_2)-Radio(i-1))*(Rigidez_Flap(i)-Rigidez_Flap(i-
1))/(Radio(i)-Radio(i-1));

                 Edge_Stiffness(cont)=Rigidez_Edge(i-
1)+(Radio_Actuador(cont_2)-Radio(i-1))*(Rigidez_Edge(i)-Rigidez_Edge(i-
1))/(Radio(i)-Radio(i-1));

                 Cuerda(cont)=Cuerda_Pala(i-1)+(Radio_Actuador(cont_2)-
Radio(i-1))*(Cuerda_Pala(i)-Cuerda_Pala(i-1))/(Radio(i)-Radio(i-1));

                 Espesor(cont)=Espesor_Pala(i-1)+(Radio_Actuador(cont_2)-
Radio(i-1))*(Espesor_Pala(i)-Espesor_Pala(i-1))/(Radio(i)-Radio(i-1));

                 Concentrated_Mass(cont)=Masa_Actuador(cont_2);

                 cont=cont+1;

                 cont_2=cont_2+1;

               end;

        end;

        if cont_1<=length(Radio_MasaMuerta)

            if Radio_MasaMuerta(cont_1)==Radio(i)

               Radius(cont)=Radio(i);

               Linear_Mass(cont)=Masa_Lineal(i);

               Flap_Stiffness(cont)=Rigidez_Flap(i);

               Edge_Stiffness(cont)=Rigidez_Edge(i);

               Cuerda(cont)=Cuerda_Pala(i);

               Espesor(cont)=Espesor_Pala(i);

               Concentrated_Mass(cont)=Masa_Muerta(cont_1);

               cont=cont+1;

               cont_1=cont_1+1;

               cepo=0;

            end;
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
        end;
        if cont_2<=length(Radio_Actuador)
              if Radio_Actuador(cont_2)==Radio(i)
                 Radius(cont)=Radio(i);
                 Linear_Mass(cont)=Masa_Lineal(i);
                 Flap_Stiffness(cont)=Rigidez_Flap(i);
                 Edge_Stiffness(cont)=Rigidez_Edge(i);
                 Cuerda(cont)=Cuerda_Pala(i);
                 Espesor(cont)=Espesor_Pala(i);
                 Concentrated_Mass(cont)=Masa_Actuador(cont_2);
                 cont=cont+1;
                 cont_2=cont_2+1;
                 cepo=0;
              end;
        end;
        if cepo==1;
            Radius(cont)=Radio(i);
            Linear_Mass(cont)=Masa_Lineal(i);
            Flap_Stiffness(cont)=Rigidez_Flap(i);
            Edge_Stiffness(cont)=Rigidez_Edge(i);
            Cuerda(cont)=Cuerda_Pala(i);
            Espesor(cont)=Espesor_Pala(i);
            Concentrated_Mass(cont)=0;
            cont=cont+1;
        end;

    elseif cont_2<=length(Radio_Actuador)
        if
(Radio_Actuador(cont_2)<Radio(i))&&(Radio_Actuador(cont_2)>Radio(i-1))
            Radius(cont)=Radio_Actuador(cont_2);
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
            Linear_Mass(cont)=Masa_Lineal(i-1)+(Radio_Actuador(cont_2)-
Radio(i-1))*(Masa_Lineal(i)-Masa_Lineal(i-1))/(Radio(i)-Radio(i-1));

            Flap_Stiffness(cont)=Rigidez_Flap(i-1)+(Radio_Actuador(cont_2)-
Radio(i-1))*(Rigidez_Flap(i)-Rigidez_Flap(i-1))/(Radio(i)-Radio(i-1));

            Edge_Stiffness(cont)=Rigidez_Edge(i-1)+(Radio_Actuador(cont_2)-
Radio(i-1))*(Rigidez_Edge(i)-Rigidez_Edge(i-1))/(Radio(i)-Radio(i-1));

            Cuerda(cont)=Cuerda_Pala(i-1)+(Radio_Actuador(cont_2)-Radio(i-
1))*(Cuerda_Pala(i)-Cuerda_Pala(i-1))/(Radio(i)-Radio(i-1));

            Espesor(cont)=Espesor_Pala(i-1)+(Radio_Actuador(cont_2)-
Radio(i-1))*(Espesor_Pala(i)-Espesor_Pala(i-1))/(Radio(i)-Radio(i-1));

            Concentrated_Mass(cont)=Masa_Actuador(cont_2);

            cont=cont+1;

            cont_2=cont_2+1;

        elseif Radio_Actuador(cont_2)==Radio(i)

               Radius(cont)=Radio(i);

               Linear_Mass(cont)=Masa_Lineal(i);

               Flap_Stiffness(cont)=Rigidez_Flap(i);

               Edge_Stiffness(cont)=Rigidez_Edge(i);

               Cuerda(cont)=Cuerda_Pala(i);

               Espesor(cont)=Espesor_Pala(i);

               Concentrated_Mass(cont)=Masa_Actuador(cont_2);

               cont=cont+1;

               cont_2=cont_2+1;

        else

            Radius(cont)=Radio(i);

            Linear_Mass(cont)=Masa_Lineal(i);

            Flap_Stiffness(cont)=Rigidez_Flap(i);

            Edge_Stiffness(cont)=Rigidez_Edge(i);

            Cuerda(cont)=Cuerda_Pala(i);

            Espesor(cont)=Espesor_Pala(i);

            Concentrated_Mass(cont)=0;

            cont=cont+1;

        end;
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
    else
          Radius(cont)=Radio(i);
          Linear_Mass(cont)=Masa_Lineal(i);
          Flap_Stiffness(cont)=Rigidez_Flap(i);
          Edge_Stiffness(cont)=Rigidez_Edge(i);
          Cuerda(cont)=Cuerda_Pala(i);
          Espesor(cont)=Espesor_Pala(i);
          Concentrated_Mass(cont)=0;
          cont=cont+1;
    end;

end;
close Actuadores_Masas_Muertas;



% --- Executes when entered data in editable cell(s) in uitable1.
function uitable1_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to uitable1 (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%   Indices: row and column indices of the cell(s) edited
%   PreviousData: previous data for the cell(s) edited
%   EditData: string(s) entered by the user
%   NewData: EditData or its converted form set on the Data property. Empty
if Data was not changed
%   Error: error string when failed to convert EditData to appropriate
value for Data
% handles    structure with handles and user data (see GUIDATA)
global Radio_Actuador1 Masa_Actuador1 Masa_Osciladora1
%Leo en una matriz valores en tabla de actuadores
AM=get(hObject,'data');
NewAM=str2double(AM);
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
Radio_Actuador1=NewAM(:,1);

Masa_Actuador1=NewAM(:,2);

Masa_Osciladora1=NewAM(:,3);



% --- Executes when entered data in editable cell(s) in uitable2.

function uitable2_CellEditCallback(hObject, eventdata, handles)

% hObject    handle to uitable2 (see GCBO)

% eventdata  structure with the following fields (see UITABLE)

%   Indices: row and column indices of the cell(s) edited

%   PreviousData: previous data for the cell(s) edited

%   EditData: string(s) entered by the user

%   NewData: EditData or its converted form set on the Data property. Empty
if Data was not changed

%   Error: error string when failed to convert EditData to appropriate
value for Data

% handles    structure with handles and user data (see GUIDATA)

global Radio_MasaMuerta1 Masa_Muerta1

%Leo en una matriz valores en tabla de masas muertas

MM=get(hObject,'data');

NewMM=str2double(MM);

Radio_MasaMuerta1=NewMM(:,1);

Masa_Muerta1=NewMM(:,2);
```

B.3 ANÁLISIS MODAL

```
function varargout = Analisis_Modal(varargin)

% ANALISIS_MODAL M-file for Analisis_Modal.fig

%      ANALISIS_MODAL, by itself, creates a new ANALISIS_MODAL or raises
the existing

%      singleton*.

%
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
%      H = ANALISIS_MODAL returns the handle to a new ANALISIS_MODAL or the
handle to

%      the existing singleton*.

%

%      ANALISIS_MODAL('CALLBACK',hObject,eventData,handles,...) calls the
local

%      function named CALLBACK in ANALISIS_MODAL.M with the given input
arguments.

%

%      ANALISIS_MODAL('Property','Value',...) creates a new ANALISIS_MODAL
or raises the

%      existing singleton*.  Starting from the left, property value pairs
are

%      applied to the GUI before Analisis_Modal_OpeningFcn gets called.  An

%      unrecognized property name or invalid value makes property
application

%      stop.  All inputs are passed to Analisis_Modal_OpeningFcn via
varargin.

%

%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one

%      instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES


% Edit the above text to modify the response to help Analisis_Modal


% Last Modified by GUIDE v2.5 20-Nov-2010 21:40:02


% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;

gui_State = struct('gui_Name',       mfilename, ...

                   'gui_Singleton',  gui_Singleton, ...

                   'gui_OpeningFcn', @Analisis_Modal_OpeningFcn, ...
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
                   'gui_OutputFcn',  @Analisis_Modal_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Analisis_Modal is made visible.
function Analisis_Modal_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Analisis_Modal (see VARARGIN)
global Rigidez Edge_Stiffness Espesor Cuerda1
a=imread('fatiguefig1.jpg');
image(a)
axis off
% Choose default command line output for Analisis_Modal
handles.output = hObject;

% Update handles structure
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
guidata(hObject, handles);

Rigidez=Edge_Stiffness;

Cuerda1=Espesor;



% UIWAIT makes Analisis_Modal wait for user response (see UIRESUME)

% uiwait(handles.figure1);



% --- Outputs from this function are returned to the command line.

function varargout = Analisis_Modal_OutputFcn(hObject, eventdata, handles)

% varargout  cell array for returning output args (see VARARGOUT);

% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)



% Get default command line output from handles structure

varargout{1} = handles.output;



% --- Executes on button press in pushbutton1.

function pushbutton1_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton1 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

global Radius Rigidez Flap_Stiffness  Edge_Stiffness D Mb Linear_Mass
Cuerda1 Masa_Osciladora Radio_Actuador Oscilator_Mass Concentrated_Mass Kb
Inversa_Masa Chord fb L w n Herze

%Se lee el numero de elementos introducido por el usuario

n=str2double(get(handles.number_elements,'String'));
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
Length=Radius(length(Radius))-Radius(1);

L=Length/n;

%Asigno propiedades a los elementos

D=zeros(n,1);

m=zeros(n,1);

chord=zeros(n,1);

% Se pasan los radios a distancias a raiz

Radio_Root=Radius(1);

for i=1:length(Radius)

    Radius(i)=Radius(i)-Radio_Root;

end;

for i=1:length(Radio_Actuador)

    Radio_Actuador(i)=Radio_Actuador(i)-Radio_Root;

end;

% Interpolo las rigideces, la masa y la cuerda o espesor para cada uno de

% los elementos

nest=1;

i=1;

    while (nest<length(Radius)) && (i<=n)

        if ((L*(i-1)+L/2)>Radius(nest))&&((L*(i-1)+L/2)<Radius(nest+1))

            D(i)=Rigidez(nest)+(Rigidez(nest+1)-
Rigidez(nest))/(Radius(nest+1)-Radius(nest))*((L*(i-1)+L/2)-Radius(nest));

            m(i)=Linear_Mass(nest)+(Linear_Mass(nest+1)-
Linear_Mass(nest))/(Radius(nest+1)-Radius(nest))*((L*(i-1)+L/2)-
Radius(nest));

            Chord(i)=Cuerda1(nest)+(Cuerda1(nest+1)-
Cuerda1(nest))/(Radius(nest+1)-Radius(nest))*((L*(i-1)+L/2)-Radius(nest));

            i=i+1;

        elseif (L/n*(i-1)+L/2)==Radius(nest+1)

            D(i)=Rigidez(nest);

            m(i)=Linear_Mass(nest);

            Chord(i)=Cuerda1(nest);
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
            i=i+1;
        else
            nest=nest+1;
        end;
end;
% Interpolo el vector de masas concentradas generado por la masa de los
% actuadores y las masas muertas, para cada uno de los nodos-.
C_M=zeros(n+1,1);
contador=1;
for i=1:(n+1)
    if (L*(i-1)<Radius(contador))&&(L*(i)>Radius(contador))
        C_M(i)=C_M(i)+Concentrated_Mass(contador)*(L*i-Radius(contador))/L;
        C_M(i+1)=C_M(i+1)+Concentrated_Mass(contador)*(Radius(contador)-
L*(i-1))/L;
        contador=contador+1;
    elseif (L*(i-1))==Radius(contador);
        C_M(i)=C_M(i)+Concentrated_Mass(contador);
        contador=contador+1;
    end;
end;
% Incorporo a la masa de los elementos, la masa de los actuadores y las
% masas muertas
for i=1:(n)
    m1(i)=m(i)+(C_M(i)/2+C_M(i+1)/2)/L;
end;



%Genero Matriz de Rigidez
K=sparse(2*n+2,2*n+2);
a=[12/L^3 6/L^2 -12/L^3 6/L^2;6/L^2 4/L -6/L^2 2/L;-12/L^3 -6/L^2 12/L^3 -
6/L^2;6/L^2 2/L -6/L^2 4/L];
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
K(1:4,1:4)=D(1)*a;

for i=2:n

   K((2*i-1):(2*i+2),(2*i-1):(2*i+2))=K((2*i-1):(2*i+2),(2*i-
1):(2*i+2))+D(i)*a;

end;

% Genero matriz de masa concentrada (lumped mass matrix).

%M_1element=[156*L/420 22*L^2/420 54*L/420 -13*L^2/420;22*L^2/420 4*L^3/420
13*L^2/420 -3*L^3/420;54*L/420 13*L^2/420 156*L/420 -22*L^2/420;-13*L^2/420
-3*L^3/420 -22*L^2/420 4*L^3/420];

M_1element=[0.5*L 0 0 0;0 1/420*L^3 0 0;0 0 0.5*L 0;0 0 0 1/420*L^3];

M=sparse(2*n+2,2*n+2);

M(1:4,1:4)=M(1:4,1:4)+m1(1)*M_1element;

for i=2:n

    M((2*i-1):(2*i+2),(2*i-1):(2*i+2))=M((2*i-1):(2*i+2),(2*i-
1):(2*i+2))+m1(i)*M_1element;

end;

% Impongo condiciones de contorno (empotramiento), a matriz de masas y a la

% de rigidez.

Kb=sparse(K(3:(2*n+2),3:(2*n+2)));

Mb=sparse(M(3:(2*n+2),3:(2*n+2)));

% Calculo la frecuencia natural

Inversa_Masa=speye(2*n)/Mb;

nat_frequ=eig(full(Inversa_Masa*Kb));

contador_3=1;

for i=1:(2*n)

    if (nat_frequ(i)>0.0000001)&&(nat_frequ(i)<100000)

        natural_frequency(contador_3)=nat_frequ(i);

        contador_3=contador_3+1;

    end;

end;

minimo=natural_frequency(1);

for i=1:(contador_3-1)
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
    if natural_frequency(i)<minimo
        minimo=natural_frequency(i);
    end;
end;
w=(minimo)^0.5;
Herze=w/(2*pi);
% Muestro en pantalla la frecuencia natrural tanto en hercios como radianes
set(handles.Radians,'String',w);
set(handles.Herze,'String',Herze);
% Genero vector de Fuerzas, debido al peso propio de la pala y de los
% actuadores
f=zeros(2*n+2,1);
f(1)=f(1)+m(1)*9.81*L/2;
for i=2:((2*n+2)/2-1)
    f(2*i-1)=f(2*i-1)+9.81*L*(m(i-1)+m(i))/2;
end;
f(2*n+1)=9.81*m(n)*L/2;
for i=1:(n+1)
    f(2*i-1)=f(2*i-1)+C_M(i)*9.81;
end;
% Genero el vector de masa oscilatoria, que define ésta en cada uno de
% los nodos
contador_1=1;
Oscilator_Mass=zeros(n+1,1);
for i=1:(n+1)
    if contador_1<=length(Radio_Actuador)
        if ((L*(i-
1))<Radio_Actuador(contador_1))&&(L*i>Radio_Actuador(contador_1))

Oscilator_Mass(i)=Oscilator_Mass(i)+Masa_Osciladora(contador_1)*(L*i-
Radio_Actuador(contador_1))/L;
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
Oscilator_Mass(i+1)=Oscilator_Mass(i+1)+Masa_Osciladora(contador_1)*(Radio_
Actuador(contador_1)-L*(i-1))/L;

            contador_1=contador_1+1;

        elseif L*(i-1)==Radio_Actuador(contador_1)


Oscilator_Mass(i)=Oscilator_Mass(i)+Masa_Osciladora(contador_1);

            contador_1=contador_1+1;

        end;

    end;

end;


% Impongo las condiciones de contorno al vector de fuerzas


fb=f(3:(2*n+2));




function Radians_Callback(hObject, eventdata, handles)

% hObject    handle to Radians (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of Radians as text

%        str2double(get(hObject,'String')) returns contents of Radians as a
double




% --- Executes during object creation, after setting all properties.

function Radians_CreateFcn(hObject, eventdata, handles)

% hObject    handle to Radians (see GCBO)
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function Herze_Callback(hObject, eventdata, handles)
% hObject    handle to Herze (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Herze as text
%        str2double(get(hObject,'String')) returns contents of Herze as a
double



% --- Executes during object creation, after setting all properties.
function Herze_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Herze (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

    set(hObject,'BackgroundColor','white');

end
```

```
function number_elements_Callback(hObject, eventdata, handles)

% hObject    handle to number_elements (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of number_elements as text

%        str2double(get(hObject,'String')) returns contents of
number_elements as a double




% --- Executes during object creation, after setting all properties.

function number_elements_CreateFcn(hObject, eventdata, handles)

% hObject    handle to number_elements (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.

%       See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

    set(hObject,'BackgroundColor','white');

end
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
% --- Executes on button press in Accept.
function Accept_Callback(hObject, eventdata, handles)
% hObject    handle to Accept (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Close Analisis_Modal



% --- Executes when selected object is changed in uipanel2.
function uipanel2_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uipanel2
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was
selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)
global Flap_Stiffness Edge_Stiffness Rigidez Cuerda Espesor Cuerda1
if (hObject==handles.flap_test)
Rigidez=Flap_Stiffness;
Cuerda1=Cuerda;
else
Rigidez=Edge_Stiffness;
Cuerda1=Espesor;
end;
```

B.4 RKN

```
function varargout = RKN(varargin)
% RKN M-file for RKN.fig
%      RKN, by itself, creates a new RKN or raises the existing
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
%      singleton*.
%
%      H = RKN returns the handle to a new RKN or the handle to
%      the existing singleton*.
%
%      RKN('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in RKN.M with the given input arguments.
%
%      RKN('Property','Value',...) creates a new RKN or raises the
%      existing singleton*.  Starting from the left, property value pairs
are
%      applied to the GUI before RKN_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to RKN_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES


% Edit the above text to modify the response to help RKN


% Last Modified by GUIDE v2.5 21-Nov-2010 23:42:19


% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @RKN_OpeningFcn, ...
                   'gui_OutputFcn',  @RKN_OutputFcn, ...
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT



% --- Executes just before RKN is made visible.
function RKN_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to RKN (see VARARGIN)
a=imread('fatiguefig1.jpg');
image(a)
axis off
% Choose default command line output for RKN
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
% UIWAIT makes RKN wait for user response (see UIRESUME)
% uiwait(handles.figure1);



% --- Outputs from this function are returned to the command line.
function varargout = RKN_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in Resolver.
function Resolver_Callback(hObject, eventdata, handles)
% hObject    handle to Resolver (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Kb Inversa_Masa Chord fb Oscilator_Mass p L n w Mb Radio D TargetMr
percent periodos
% Lee el número de periodos a simular, a que porcentaje frecuencia
% natural queremos simular el ensayo
periodos=str2double(get(handles.periodos,'String'));
percent=str2double(get(handles.percent,'String'));
p=str2double(get(handles.paso,'String'));
% Predefine el vector desplazamientos
u=zeros(2*n+2,1);
% Resuelve el sistema estático para hallar la deformada de la pala por
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
% su propio peso, que será tomada como condición inicial

%R=chol(Kb);

%Un=R\(R'\fb);

[Y,X,N]=lu(Kb);

bb=fb;

bbnew=N*bb;

Un=X\(Y\bbnew);

% Se definen los coeficientes del tablero de Butcher RKN

c1=(9+(33)^0.5)/12;

c2=(1+(33)^0.5)/(6+2*(33)^0.5);

a11=((9+(33)^0.5)^2)/288;

a21=-((99+19*(33)^0.5)/(4*(3+(33)^0.5)^2));

a22=a11;

beta1=(3-(33)^0.5)/(54+6*(33)^0.5);

beta2=(2*(6+(33)^0.5))/(3*(9+(33)^0.5));

b1=2/(9+(33)^0.5);

b2=(7+(33)^0.5)/(9+(33)^0.5);


step=1;

t=0;

w1=percent*w;

% Se define el vector velocidades como ceros. Será la otra condición
inicial

Vn=zeros(2*n,1);

%Se definen vectores de fuerzas de primera y segunda iteración

fb11=zeros(2*n,1);

fb12=zeros(2*n,1);

fb21=zeros(2*n,1);

fb22=zeros(2*n,1);

step=1;
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
t=0;

w1=(percent/100)*w;

u(3:(2*n+2),1)=Un;

% Se hace la factorización LU para la primera y segunda etapa del RKN

[l,T,P]=lu(speye(2*n)+(p^2)*a11*Inversa_Masa*Kb);

[Z,S,G]=lu(speye(2*n)+(p^2)*a22*Inversa_Masa*Kb);

% Comienza la resolución temporal del RKN a lo largo de los periodos

% introducidos por el usuario.

while t<=(periodos*2*pi/w1)

    for i=1:(n-1)

       if Vn(2*i-1)~=0

       fb11(2*i-1)=fb(2*i-1)-(Vn(2*i-1)/abs(Vn(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(c1*p+t));

       fb12(2*i-1)=fb(2*i-1)-(Vn(2*i-1)/abs(Vn(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(t+p*c2));

       else

       fb11(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(c1*p+t));

       fb12(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(t+p*c2));

       end;

   end;

   if Vn(2*n-1)~=0

        fb11(2*n-1)=fb(2*n-1)-(Vn(2*n-1)/abs(Vn(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c1*p));

        fb12(2*n-1)=fb(2*n-1)-(Vn(2*n-1)/abs(Vn(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c2*p));

   else

        fb11(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c1*p));

        fb12(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c2*p));
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
    end;

    b111=(Un+c1*p*Vn+(p^2)*a11*Inversa_Masa*fb11);

    b11new=P*b111;

    Q11=T\(l\b11new);

    b222=Un+c2*p*Vn+(p^2)*(a21*Inversa_Masa*(fb11-
Kb*Q11)+a22*Inversa_Masa*fb12);

    b22new=G*b222;

    Q22=S\(Z\b22new);

    V1=Vn+p*b1*Inversa_Masa*(fb11-Kb*Q11)+p*b2*Inversa_Masa*(fb12-Kb*Q22);



    for i=1:(n-1)

        if V1(2*n-1)~=0

        fb21(2*i-1)=fb(2*i-1)-(V1(2*i-1)/abs(V1(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(c1*p+t));

        fb22(2*i-1)=fb(2*i-1)-(V1(2*i-1)/abs(V1(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(t+p*c2));

        else

        fb21(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(c1*p+t));

        fb22(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(t+p*c2));

        end;

    end;

    if V1(2*n-1)~=0

         fb21(2*n-1)=fb(2*n-1)-(V1(2*n-1)/abs(V1(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c1*p));

         fb22(2*n-1)=fb(2*n-1)-(V1(2*n-1)/abs(V1(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c2*p));

    else

         fb21(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c1*p));

         fb22(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c2*p));
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
    end;

    b11=(Un+c1*p*Vn+(p^2)*a11*Inversa_Masa*fb21);

    b1new=P*b11;

    Q1=T\(l\b1new);

    b22=Un+c2*p*Vn+(p^2)*(a21*Inversa_Masa*(fb21-
Kb*Q1)+a22*Inversa_Masa*fb22);

    b2new=G*b22;

    Q2=S\(Z\b2new);

    V=Vn+p*b1*Inversa_Masa*(fb21-Kb*Q1)+p*b2*Inversa_Masa*(fb22-Kb*Q2);

    U=Un+p*Vn+(p^2)*beta1*Inversa_Masa*(fb21-
Kb*Q1)+beta2*(p^2)*Inversa_Masa*(fb22-Kb*Q2);

    Vn=V;

    Un=U;

    step=step+1;

    u(3:(2*n+2),step)=U;

    t=t+p;

end;

u(1,:)=0;

u(2,:)=0;

% Dibujamos la punta de la pala a lo largo de la simulación

x=1:1:step;

figure

plot(x,u(2*n+1,:))

ylabel('Desplazamiento punta de la pala [m]','fontsize',14);

xlabel('Pasos','fontsize',14);

% Hallo desplazamiento maximo y minimo

estab=fix((5*2*pi/w1)/p);

umaxim=u(2*n+1,step-estab);

uminim=u(2*n+1,step-estab);

for i=(step-estab):step

    if u(2*n+1,i)>umaxim
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
        maximo=i;
        umaxim=u(2*n+1,i);
    end;
    if u(2*n+1,i)<uminim
        minimo=i;
        uminim=u(2*n+1,i);
    end;
end;
% Hallo desplazamiento de rango y medio
Um=zeros(n+1,1);
Ur=zeros(n+1,1);
for i=1:(n+1)
    Um(i)=(u(2*i-1,maximo)+u(2*i-1,minimo))/2;
    Ur(i)=u(2*i-1,maximo)-u(2*i-1,minimo);
end;
% A partir de los desplazamientos de rango y los medios,por una doble
% derivación numérica, obtenemos los momentos rango y medio.
Mr=zeros(n+1,1);
Mm=zeros(n+1,1);
for i=2:(n)
    Mr(i)=((D(i-1)+D(i))/2)*(Ur(i+1)-2*Ur(i)+Ur(i-1))/(L^2);
    Mm(i)=((D(i-1)+D(i))/2)*(Um(i+1)-2*Um(i)+Um(i-1))/(L^2);
end;
Mr(1)=Mr(2)+(Mr(2)-Mr(3));
Mm(1)=Mm(2)+(Mm(2)-Mm(3));
Mr(n+1)=D(n)*(Ur(n+1)-2*Ur(n)+Ur(n-1))/(L^2);
Mm(n+1)=D(n)*(Um(n+1)-2*Um(n)+Um(n-1))/(L^2);

% Interporlo desplazamientos y momentos para los radios que tengo datos
% objetivo
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
ZRadio=zeros(length(Radio),1);
for i=1:length(Radio)
    ZRadio(i)=Radio(i)-Radio(1);
end;
Solucion=zeros(length(Radio),5);
Solucion(:,1)=Radio;
cont_4=1;
i=1;


while cont_4<=length(ZRadio)
    if ZRadio(cont_4)==L*(i-1)
        Solucion(cont_4,2)=Um(i);
        Solucion(cont_4,3)=Ur(i);
        Solucion(cont_4,4)=Mm(i);
        Solucion(cont_4,5)=Mr(i);
        cont_4=cont_4+1;
    elseif (ZRadio(cont_4)>L*(i-1))&&(ZRadio(cont_4)<L*i)
        Solucion(cont_4,2)=Um(i)+(ZRadio(cont_4)-L*(i-1))*(Um(i+1)-
Um(i))/L;
        Solucion(cont_4,3)=Ur(i)+(ZRadio(cont_4)-L*(i-1))*(Ur(i+1)-
Ur(i))/L;
        Solucion(cont_4,4)=Mm(i+1)+(L*i-ZRadio(cont_4))*(Mm(i)-Mm(i+1))/L;
        Solucion(cont_4,5)=Mr(i+1)+(L*i-ZRadio(cont_4))*(Mr(i)-Mr(i+1))/L;
        cont_4=cont_4+1;
    else
        i=i+1;
    end;
end;
Solucion(:,5)
% Dibujo en un mismo gráfico los momentos rango objetivo y los momentos de
% test
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
figure

plot(Radio,Solucion(:,5),'-ro',Radio,TargetMr,'-.b')

legend('Momentos de Ensayo','Momentos Objetivo')

ylabel('Momento [Nm]','fontsize',11);

xlabel('Radio [m]','fontsize',11);

xlswrite('Simulacion.xls',Solucion);

close RKN
```

```
% --- Executes on button press in Optimize.

function Optimize_Callback(hObject, eventdata, handles)

% hObject    handle to Optimize (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

global Kb Inversa_Masa Chord fb Oscilator_Mass p L n w Mb Radio D TargetMr
Radio_Actuador periodos percent

u=zeros(2*n+2,1);


R=chol(Kb);

Un=R\(R'\fb);


c1=(9+(33)^0.5)/12;

c2=(1+(33)^0.5)/(6+2*(33)^0.5);

a11=((9+(33)^0.5)^2)/288;

a21=-((99+19*(33)^0.5)/(4*(3+(33)^0.5)^2));

a22=a11;

beta1=(3-(33)^0.5)/(54+6*(33)^0.5);

beta2=(2*(6+(33)^0.5))/(3*(9+(33)^0.5));

b1=2/(9+(33)^0.5);
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
b2=(7+(33)^0.5)/(9+(33)^0.5);


step=1;

t=0;

w1=(percent/100)*w;

Vn=zeros(2*n,1);

fb11=zeros(2*n,1);

fb12=zeros(2*n,1);

fb21=zeros(2*n,1);

fb22=zeros(2*n,1);


u(3:(2*n+2),1)=Un;

periodos=str2double(get(handles.periodos,'String'));

percent=str2double(get(handles.percent,'String'));

p=str2double(get(handles.paso,'String'));

[l,T,P]=lu(speye(2*n)+(p^2)*a11*Inversa_Masa*Kb);

[Z,S,G]=lu(speye(2*n)+(p^2)*a22*Inversa_Masa*Kb);

% Se resuelve la primera iteración antes de meterse en el bucle optimizador

while t<=(periodos*2*pi/w1)

    for i=1:(n-1)

        if Vn(2*i-1)~=0

        fb11(2*i-1)=fb(2*i-1)-(Vn(2*i-1)/abs(Vn(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(c1*p+t));

        fb12(2*i-1)=fb(2*i-1)-(Vn(2*i-1)/abs(Vn(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(t+p*c2));

        else

        fb11(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(c1*p+t));

        fb12(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(t+p*c2));

        end;
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
   end;

   if Vn(2*n-1)~=0

        fb11(2*n-1)=fb(2*n-1)-(Vn(2*n-1)/abs(Vn(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c1*p));

        fb12(2*n-1)=fb(2*n-1)-(Vn(2*n-1)/abs(Vn(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c2*p));

   else

        fb11(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c1*p));

        fb12(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c2*p));

   end;

   b111=(Un+c1*p*Vn+(p^2)*a11*Inversa_Masa*fb11);

   b11new=P*b111;

   Q11=T\(l\b11new);

   b222=Un+c2*p*Vn+(p^2)*(a21*Inversa_Masa*(fb11-
Kb*Q11)+a22*Inversa_Masa*fb12);

   b22new=G*b222;

   Q22=S\(Z\b22new);

   V1=Vn+p*b1*Inversa_Masa*(fb11-Kb*Q11)+p*b2*Inversa_Masa*(fb12-Kb*Q22);



   for i=1:(n-1)

       if V1(2*n-1)~=0

       fb21(2*i-1)=fb(2*i-1)-(V1(2*i-1)/abs(V1(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(c1*p+t));

       fb22(2*i-1)=fb(2*i-1)-(V1(2*i-1)/abs(V1(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(t+p*c2));

       else

       fb21(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(c1*p+t));

       fb22(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(w1^2)*0.423*sin(w1*(t+p*c2));

       end;
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
    end;

    if V1(2*n-1)~=0

        fb21(2*n-1)=fb(2*n-1)-(V1(2*n-1)/abs(V1(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c1*p));

        fb22(2*n-1)=fb(2*n-1)-(V1(2*n-1)/abs(V1(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c2*p));

    else

        fb21(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c1*p));

        fb22(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(w1^2)*0.423*sin(w1*(t+c2*p));

    end;

    b11=(Un+c1*p*Vn+(p^2)*a11*Inversa_Masa*fb21);

    b1new=P*b11;

    Q1=T\(l\b1new);

    b22=Un+c2*p*Vn+(p^2)*(a21*Inversa_Masa*(fb21-
Kb*Q1)+a22*Inversa_Masa*fb22);

    b2new=G*b22;

    Q2=S\(Z\b2new);

    V=Vn+p*b1*Inversa_Masa*(fb21-Kb*Q1)+p*b2*Inversa_Masa*(fb22-Kb*Q2);

    U=Un+p*Vn+(p^2)*beta1*Inversa_Masa*(fb21-
Kb*Q1)+beta2*(p^2)*Inversa_Masa*(fb22-Kb*Q2);

    Un=U;

    Vn=V;

    step=step+1;

    u(3:(2*n+2),step)=U;

    t=t+p;

end;

u(1,:)=0;

u(2,:)=0;

u(2*n+1,:);

x=1:1:step;
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
figure
plot(x,u(2*n+1,:))
ylabel('Desplazamiento punta de pala [m]','fontsize',14);
xlabel('Pasos','fontsize',14);
% Hallo desplazamiento maximo y minimo
estab=fix((5*2*pi/w1)/p);
umaxim=u(2*n+1,step-estab);
uminim=u(2*n+1,step-estab);
for i=(step-estab):step
    if u(2*n+1,i)>umaxim
        maximo=i;
        umaxim=u(2*n+1,i);
    end;
    if u(2*n+1,i)<uminim
        minimo=i;
        uminim=u(2*n+1,i);
    end;
end;
% Obtengo el desplazamiento de rango y medio
Um=zeros(n+1,1);
Ur=zeros(n+1,1);
for i=1:(n+1)
    Um(i)=(u(2*i-1,maximo)+u(2*i-1,minimo))/2;
    Ur(i)=u(2*i-1,maximo)-u(2*i-1,minimo);
end;
% A partir de los desplazamientos de rango y los medios,por una doble
% derivación numérica, obtenemos los momentos rango y medio.
Mr=zeros(n+1,1);
Mm=zeros(n+1,1);
for i=2:(n)
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
    Mr(i)=((D(i-1)+D(i))/2)*(Ur(i+1)-2*Ur(i)+Ur(i-1))/(L^2);
    Mm(i)=((D(i-1)+D(i))/2)*(Um(i+1)-2*Um(i)+Um(i-1))/(L^2);
end;
Mr(1)=Mr(2)+(Mr(2)-Mr(3));
Mm(1)=Mm(2)+(Mm(2)-Mm(3));
Mr(n+1)=D(n)*(Ur(n+1)-2*Ur(n)+Ur(n-1))/(L^2);
Mm(n+1)=D(n)*(Um(n+1)-2*Um(n)+Um(n-1))/(L^2);

% Interporlo desplazamientos y momentos para los radios que tengo datos
% objetivo
ZRadio=zeros(length(Radio),1);
for i=1:length(Radio)
    ZRadio(i)=Radio(i)-Radio(1);
end;
Solucion=zeros(length(Radio),5);
Solucion(:,1)=Radio;
cont_4=1;
i=1;

while cont_4<=length(ZRadio)
    if ZRadio(cont_4)==L*(i-1)
        Solucion(cont_4,2)=Um(i);
        Solucion(cont_4,3)=Ur(i);
        Solucion(cont_4,4)=Mm(i);
        Solucion(cont_4,5)=Mr(i);
        cont_4=cont_4+1;
    elseif (ZRadio(cont_4)>L*(i-1))&&(ZRadio(cont_4)<L*i)
        Solucion(cont_4,2)=Um(i)+(ZRadio(cont_4)-L*(i-1))*(Um(i+1)-
Um(i))/L;
        Solucion(cont_4,3)=Ur(i)+(ZRadio(cont_4)-L*(i-1))*(Ur(i+1)-
Ur(i))/L;
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
        Solucion(cont_4,4)=Mm(i+1)+(L*i-ZRadio(cont_4))*(Mm(i)-Mm(i+1))/L;
        Solucion(cont_4,5)=Mr(i+1)+(L*i-ZRadio(cont_4))*(Mr(i)-Mr(i+1))/L;
        cont_4=cont_4+1;
    else
        i=i+1;
    end;
end;
% Dibujo en un mismo gráfico los momentos rango objetivo y los momentos de
% test
figure
plot(Radio,Solucion(:,5),'-ro',Radio,TargetMr,'-.b')
legend('Momentos de Ensayo','Momentos Objetivo')
ylabel('Momento [Nm]','fontsize',14);
xlabel('Radio [m]','fontsize',14);
% Calculo error de iteración inicial
cont_5=0;
for i=1 length(Radio)
    if Radio(i)<Radio_Actuador(length(Radio_Actuador))
        cont_5=cont_5+1;
    end;
end;
SUMA_ERRORES=0;
cuento=0;
    for i=1:cont_5
        if TargetMr(i)~=0
            SUMA_ERRORES=SUMA_ERRORES+(Solucion(i,5)/TargetMr(i))-1;
        else
            cuento=cuento+1;
        end;
    end;
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
    PROMEDIO=SUMA_ERRORES/(cont_5-cuento);
% Predefino cotas máxima y mínima de rango de frecuencias de trabajo
wsol=w1;
want=0;
wmax=0.985*w;
wmin=0.8*w;
% Se inicia el proceso iterativo optimizador
while ((wsol-want)>0.0628)&&(abs(PROMEDIO)>0.01)
    if PROMEDIO<0
        wmin=wsol;
    elseif PROMEDIO>0
        wmax=wsol;
    end;
    want=wsol;
    wsol=(wmax+wmin)/2
    u=zeros(2*n+2,1);

    R=chol(Kb);
    Un=R\(R'\fb);
    step=1;
    t=0;
    Vn=zeros(2*n,1);
    fb11=zeros(2*n,1);
    fb12=zeros(2*n,1);
    fb21=zeros(2*n,1);
    fb22=zeros(2*n,1);
    u(3:(2*n+2),1)=Un;

   %[l,T,P]=lu(speye(2*n)+(p^2)*a11*Inversa_Masa*Kb);
   %[Z,S,G]=lu(speye(2*n)+(p^2)*a22*Inversa_Masa*Kb);
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
   while t<=(periodos*2*pi/wsol)

     for i=1:(n-1)

        if Vn(2*i-1)~=0

           fb11(2*i-1)=fb(2*i-1)-(Vn(2*i-1)/abs(Vn(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(wsol^2)*0.423*sin(wsol*(c1*p+t));

           fb12(2*i-1)=fb(2*i-1)-(Vn(2*i-1)/abs(Vn(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(wsol^2)*0.423*sin(wsol*(t+p*c2));

        else

           fb11(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(wsol^2)*0.423*sin(wsol*(c1*p+t));

           fb12(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(wsol^2)*0.423*sin(wsol*(t+p*c2));

        end;

     end;

     if Vn(2*n-1)~=0

         fb11(2*n-1)=fb(2*n-1)-(Vn(2*n-1)/abs(Vn(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(wsol^2)*0.423*sin(wsol*(t+c1*p));

         fb12(2*n-1)=fb(2*n-1)-(Vn(2*n-1)/abs(Vn(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(wsol^2)*0.423*sin(wsol*(t+c2*p));

     else

         fb11(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(wsol^2)*0.423*sin(wsol*(t+c1*p));

         fb12(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(wsol^2)*0.423*sin(wsol*(t+c2*p));

     end;

     b111=(Un+c1*p*Vn+(p^2)*a11*Inversa_Masa*fb11);

     b11new=P*b111;

     Q11=T\(l\b11new);

     b222=Un+c2*p*Vn+(p^2)*(a21*Inversa_Masa*(fb11-
Kb*Q11)+a22*Inversa_Masa*fb12);

     b22new=G*b222;

     Q22=S\(Z\b22new);

     V1=Vn+p*b1*Inversa_Masa*(fb11-Kb*Q11)+p*b2*Inversa_Masa*(fb12-Kb*Q22);
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
   for i=1:(n-1)

       if V1(2*n-1)~=0

       fb21(2*i-1)=fb(2*i-1)-(V1(2*i-1)/abs(V1(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(wsol^2)*0.423*sin(wsol*(c1*p+t));

       fb22(2*i-1)=fb(2*i-1)-(V1(2*i-1)/abs(V1(2*i-
1)))*(L/4)*1.255*((Vn(2*i-1))^2)*(Chord(i)+Chord(i+1))-
Oscilator_Mass(i+1)*(wsol^2)*0.423*sin(wsol*(t+p*c2));

       else

       fb21(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(wsol^2)*0.423*sin(wsol*(c1*p+t));

       fb22(2*i-1)=fb(2*i-1)-
Oscilator_Mass(i+1)*(wsol^2)*0.423*sin(wsol*(t+p*c2));

       end;

   end;

   if V1(2*n-1)~=0

        fb21(2*n-1)=fb(2*n-1)-(V1(2*n-1)/abs(V1(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(wsol^2)*0.423*sin(wsol*(t+c1*p));

        fb22(2*n-1)=fb(2*n-1)-(V1(2*n-1)/abs(V1(2*n-
1)))*(L/4)*1.255*((Vn(2*n-1))^2)*(Chord(n))-
Oscilator_Mass(n+1)*(wsol^2)*0.423*sin(wsol*(t+c2*p));

   else

        fb21(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(wsol^2)*0.423*sin(wsol*(t+c1*p));

        fb22(2*n-1)=fb(2*n-1)-
Oscilator_Mass(n+1)*(wsol^2)*0.423*sin(wsol*(t+c2*p));

   end;

   b11=(Un+c1*p*Vn+(p^2)*a11*Inversa_Masa*fb21);

   b1new=P*b11;

   Q1=T\(l\b1new);

   b22=Un+c2*p*Vn+(p^2)*(a21*Inversa_Masa*(fb21-
Kb*Q1)+a22*Inversa_Masa*fb22);

   b2new=G*b22;

   Q2=S\(Z\b2new);

   V=Vn+p*b1*Inversa_Masa*(fb21-Kb*Q1)+p*b2*Inversa_Masa*(fb22-Kb*Q2);
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
   U=Un+p*Vn+(p^2)*beta1*Inversa_Masa*(fb21-
Kb*Q1)+beta2*(p^2)*Inversa_Masa*(fb22-Kb*Q2);

   Vn=V;

   Un=U;

   step=step+1;

   u(3:(2*n+2),step)=U;

   t=t+p;

   end;

   u(1,:)=0;

   u(2,:)=0;

   u(2*n+1,:);

   x=1:1:step;

   figure

   plot(x,u(2*n+1,:))

   ylabel('Desplazamiento punta de pala [m]','fontsize',14);

   xlabel('Pasos','fontsize',14);

% Hallo desplazamiento maximo y minimo

   estab=fix((5*2*pi/wsol)/p);

   umaxim=u(2*n+1,step-estab);

   uminim=u(2*n+1,step-estab);

   for i=(step-estab):step

        if u(2*n+1,i)>umaxim

            maximo=i;

            umaxim=u(2*n+1,i);

        end;

        if u(2*n+1,i)<uminim

            minimo=i;

            uminim=u(2*n+1,i);

        end;

   end;

% Hallo desplazamiento de rango y medio
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
    Um=zeros(n+1,1);
    Ur=zeros(n+1,1);
    for i=1:(n+1)
        Um(i)=(u(2*i-1,maximo)+u(2*i-1,minimo))/2;
        Ur(i)=u(2*i-1,maximo)-u(2*i-1,minimo);
    end;
% A partir de los desplazamientos de rango y los medios,por una doble
% derivación numérica, obtenemos los momentos rango y medio.
    Mr=zeros(n+1,1);
    Mm=zeros(n+1,1);
    for i=2:(n)
        Mr(i)=((D(i-1)+D(i))/2)*(Ur(i+1)-2*Ur(i)+Ur(i-1))/(L^2);
        Mm(i)=((D(i-1)+D(i))/2)*(Um(i+1)-2*Um(i)+Um(i-1))/(L^2);
    end;
    Mr(1)=Mr(2)+(Mr(2)-Mr(3));
    Mm(1)=Mm(2)+(Mm(2)-Mm(3));
    Mr(n+1)=D(n)*(Ur(n+1)-2*Ur(n)+Ur(n-1))/(L^2);
    Mm(n+1)=D(n)*(Um(n+1)-2*Um(n)+Um(n-1))/(L^2);

% Interporlo desplazamientos y momentos para los radios que tengo datos
% objetivo
    ZRadio=zeros(length(Radio),1);
    for i=1:length(Radio)
        ZRadio(i)=Radio(i)-Radio(1);
    end;
    Solucion=zeros(length(Radio),5);
    Solucion(:,1)=Radio;
    cont_4=1;
    i=1;
    while cont_4<=length(ZRadio)
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
        if ZRadio(cont_4)==L*(i-1)
            Solucion(cont_4,2)=Um(i);
            Solucion(cont_4,3)=Ur(i);
            Solucion(cont_4,4)=Mm(i);
            Solucion(cont_4,5)=Mr(i);
            cont_4=cont_4+1;
        elseif (ZRadio(cont_4)>L*(i-1))&&(ZRadio(cont_4)<L*i)
            Solucion(cont_4,2)=Um(i)+(ZRadio(cont_4)-L*(i-1))*(Um(i+1)-
Um(i))/L;
            Solucion(cont_4,3)=Ur(i)+(ZRadio(cont_4)-L*(i-1))*(Ur(i+1)-
Ur(i))/L;
            Solucion(cont_4,4)=Mm(i+1)+(L*i-ZRadio(cont_4))*(Mm(i)-
Mm(i+1))/L;
            Solucion(cont_4,5)=Mr(i+1)+(L*i-ZRadio(cont_4))*(Mr(i)-
Mr(i+1))/L;
            cont_4=cont_4+1;
        else
            i=i+1;
        end;
    end;
SUMA_ERRORES=0;
cuento=0;
    for i=1:cont_5
        if TargetMr(i)~=0
            SUMA_ERRORES=SUMA_ERRORES+(Solucion(i,5)/TargetMr(i))-1;
        else
            cuento=cuento+1;
        end;
    end;
    PROMEDIO=SUMA_ERRORES/(cont_5-cuento);
    figure
plot(Radio,Solucion(:,5),'-ro',Radio,TargetMr,'-.b')
```

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

```
legend('Momentos de Ensayo','Momentos Objetivo')
ylabel('Momento [Nm]','fontsize',14);
xlabel('Radio [m]','fontsize',14);
end;
% Dibujo en un mismo gráfico los momentos rango objetivo y los momentos de
% test
figure
plot(Radio,Solucion(:,5),'-ro',Radio,TargetMr,'-.b')
legend('Momentos de Ensayo','Momentos Objetivo')
ylabel('Momento [Nm]','fontsize',14);
xlabel('Radio [m]','fontsize',14);
xlswrite('Simulacion.xls',Solucion);
wsol

close RKN



% --- Executes on button press in target_loads.
function target_loads_Callback(hObject, eventdata, handles)
% hObject    handle to target_loads (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Lee el archivo con las cargas objetivo
[FileName Path]=uigetfile({'*.m;*.mdl'}, 'Escoger');
global TargetMr TargetMm
TM=xlsread(FileName);
TargetMr=TM(:,2);
TargetMm=TM(:,3)
```


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

SIMULACIÓN DE ENSAYOS A FATIGA DE PALAS DE AEROGENERADORES

Mikel Mendia Villaamil

Ingeniería Industrial

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

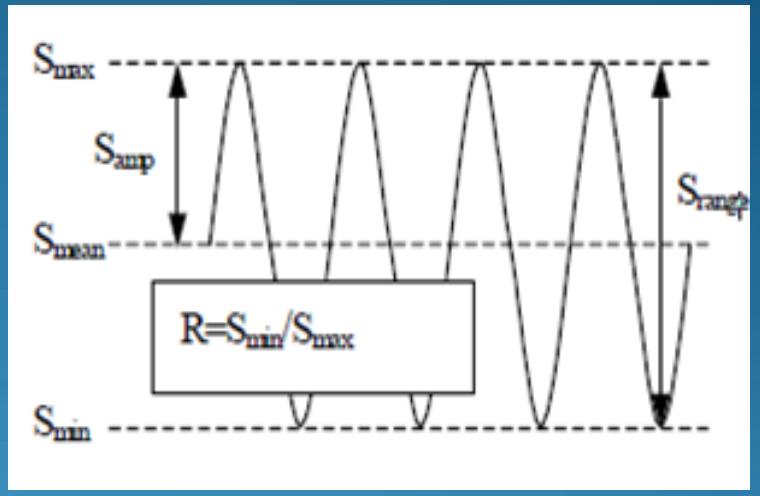
1. ENERGÍA EÓLICA

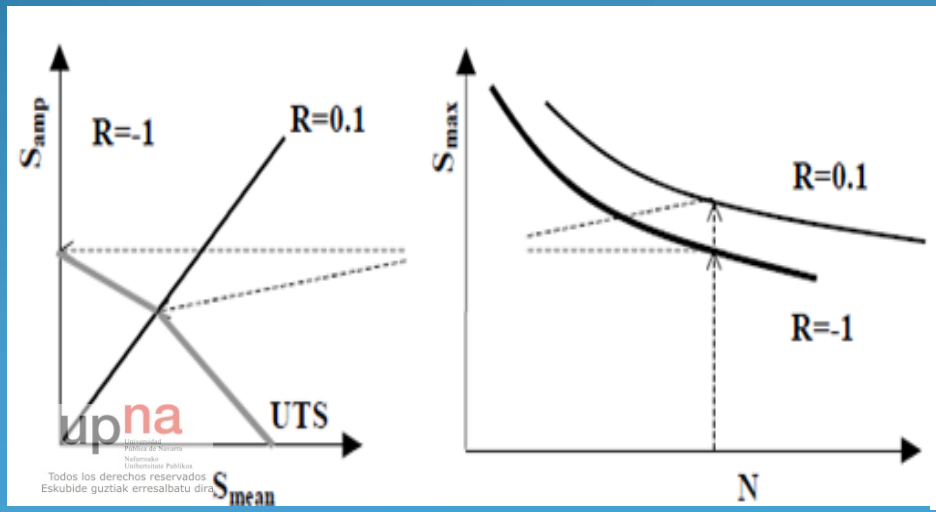
- Primera forma de aprovechamiento de la energía eólica en navegación.
- Molienda y bombeo de agua.
- Causas para el gran desarrollo en la actualidad:
  I. Incremento del precio de combustibles fósiles.
  II. Combustibles fósiles recurso limitado.
  III. Contaminación.

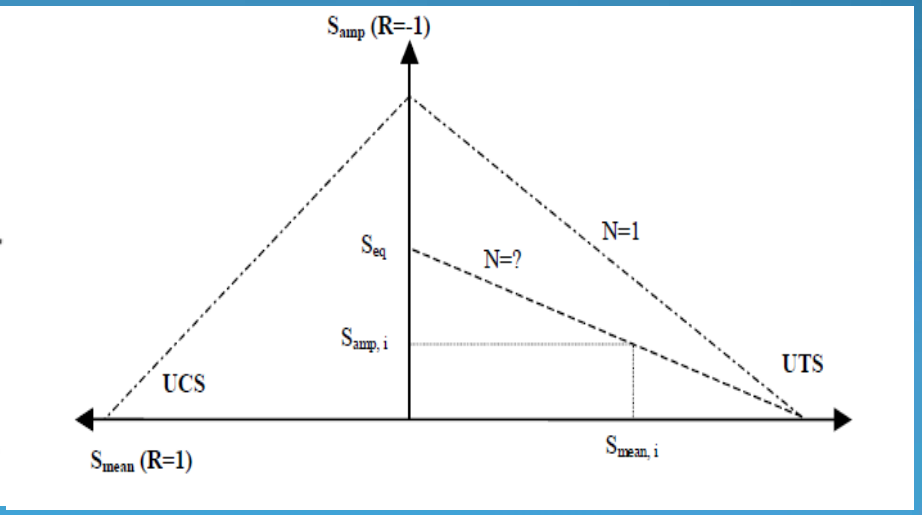
upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresaibatu dira

2. FATIGA

- Cargas cíclicas en el tiempo (S_{amp} y S_{min}).
- Critica en palas de aerogeneradores debido a:
  I. Número ciclos (20 años☐2e08 ciclos).
  II. Variabilidad de las cargas.
  III. Baja predictibilidad de las cargas.
- Diagramas S-N (logN=a +mS).
- Diagramas CLD



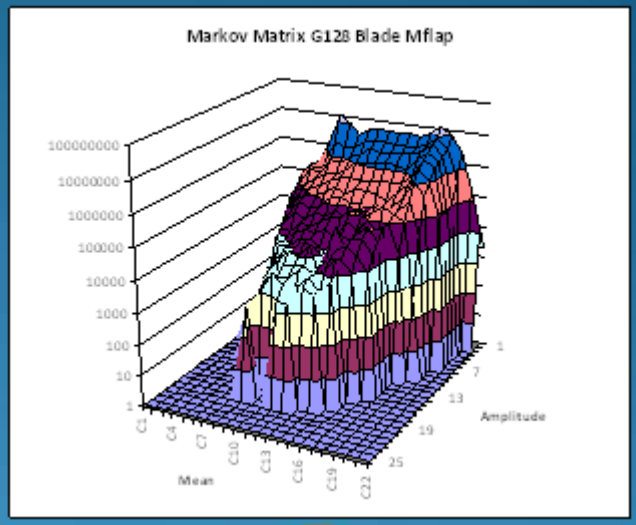


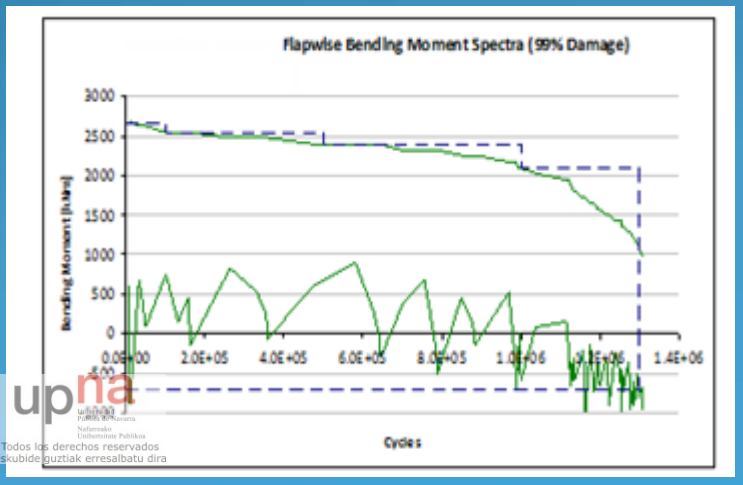


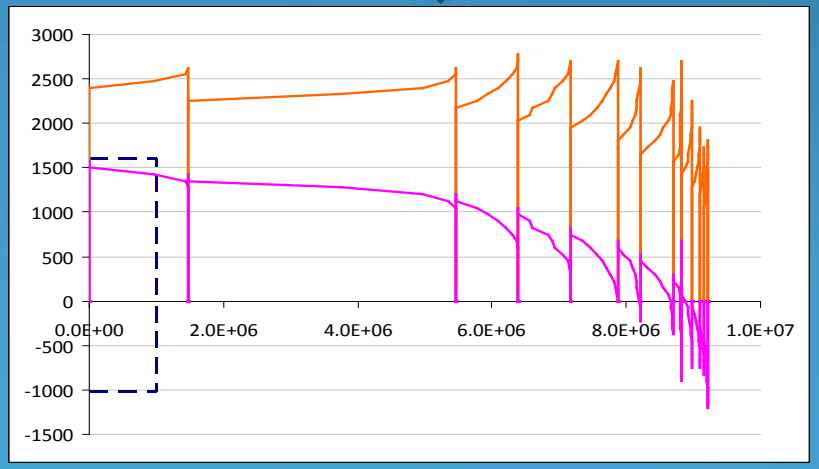
upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

3. OBTENCIÓN DE CARGAS

- Suma de Miner. $D = \sum_i \frac{n_i}{N_i}$
- Recuento de ciclos (*Rainflowcounting*…).
- Filtrado de ciclos.
- Selección de niveles de carga a ensayar.







upna
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

4. ENSAYOS A FATIGA

Para alcanzar nivel de carga objetivo (Momentos medio y de rango):

- Primera frecuencia natural de vibración.
- Actuadores hidráunlicos o de masa rotatoria.
- Masas muertas.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

5. ECUACIÓN BIARMÓNICA DE ONDAS

$$\frac{\partial^2}{\partial x^2}\left(D(x)\cdot\frac{\partial^2 u(x,t)}{\partial x^2}\right)+m(x)\cdot\frac{\partial^2 u(x,t)}{\partial t^2}=f(x,t)$$

Donde:

- D (x) [Nm^2] es la rigidez de la viga (E(x) I(x)).
- u (x,t) [m] es el desplazamiento en la dirección de la carga.
- x [m] es la dimensión longitudinal de la pala.
- t [s] es el tiempo.
- m (x) [kg] es la masa por unidad de longitud de la pala.
- f(x,t) [N/m] es la fuerza por unidad de longitud a lo largo del elemento.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

6. DISCRETIZACIÓN ESPACIAL
6.1 CONDICIONES DE CONTORNO

- Desplazamiento en empotramiento nulo ⟹ $u(0,t)=0$
- Giro en empotramiento nulo ⟹ $\frac{\partial u}{\partial x}(0,t)=0$
- Momento flector en extremo libre nulo ⟹ $\frac{\partial^2 u}{\partial x^2}(L,t)=0$
- Cortante en extremo libre nulo ⟹ $\frac{\partial^3 u}{\partial x^3}(L,t)=0$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

6. DISCRETIZACIÓN ESPACIAL
6.2 FORMULACIÓN VARIACIONAL

Consideramos el espacio $H^2(0,L) := \{u \in L^2(0,L) | u', u'' \in L^2(0,L)\}$

y el subespacio $V := \{u \in H^2(0,L) |\, u(0) = u'(0) = 0\}$

$$\frac{\partial^2}{\partial x^2}\left(D(x) \cdot \frac{\partial^2 u(x,t)}{\partial x^2}\right) + m(x) \cdot \frac{\partial^2 u(x)}{\partial t^2} = f(x,t)$$

$$\int_0^L \frac{\partial^2}{\partial x^2}\left(D(x) \cdot \frac{\partial^2 u(x,t)}{\partial x^2}\right) \cdot \mathrm{v} \cdot \mathrm{dx} + \int_0^L \left(m(x) \cdot \frac{\partial^2 u(x,t)}{\partial t^2}\right) \cdot \mathrm{v} \cdot \mathrm{dx} = \int_0^L f(x,t) \cdot \mathrm{v} \cdot \mathrm{dx}$$

$$\int_0^L \left(D(x) \cdot \frac{\partial^2 u(x,t)}{\partial x^2} \cdot \frac{\partial^2 v(x)}{\partial x^2}\right) \cdot \mathrm{dx} + \int_0^L m(x) \cdot \frac{\partial^2 u(x,t)}{\partial x^2} \cdot \mathrm{v} \cdot \mathrm{dx} = \int_0^L f(x,t) \cdot \mathrm{v} \cdot \mathrm{dx}$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

6. DISCRETIZACIÓN ESPACIAL
6.3 ELEMENTOS DE HERMITE

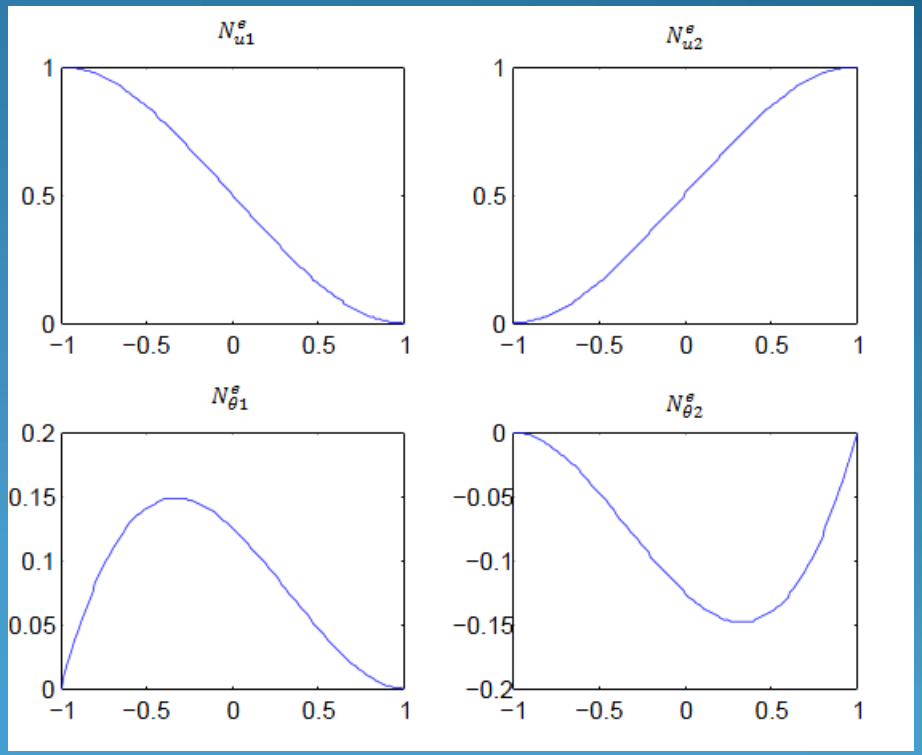
- Se emplean elementos de Hermite 1D, teniendo las siguientes funciones de forma:

$$N^e_{u1} = \frac{1}{4}\cdot(\xi-1)^2\cdot(\xi+2)$$

$$N^e_{u2} = -\frac{1}{4}\cdot(\xi+1)^2\cdot(\xi-2)$$

$$N^e_{\theta 1} = \frac{1}{8}\cdot L^e\cdot(\xi+1)\cdot(\xi-1)^2$$

$$N^e_{\theta 2} = \frac{1}{8}\cdot L^e\cdot(\xi+1)^2\cdot(\xi-1)$$



$$\{\hat{u}^e\} = \{\hat{u}(x,t)|_e\} = \{N^e(x)\}\cdot\{u^e(t)\}\ ,$$

$$\{\hat{u}^e\} = [N^e_{u1}(x)\quad N^e_{\theta 1}(x)\quad N^e_{u2}(x)\quad N^e_{\theta 2}(x)]\cdot\begin{bmatrix} u1(t)\\ \theta 1(t)\\ u2(t)\\ \theta 2(t)\end{bmatrix} =$$

$$= u^e_1(t)\cdot N^e_{u1}(x) + u^e_2(t)\cdot N^e_{u2}(x) + \left(\frac{du^e_1}{dx}\right)(t)\cdot N^e_{\theta 1}(x) + \left(\frac{du^e_2}{dx}\right)(x)\cdot N^e_{\theta 2}(x)$$

upna
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

6. DISCRETIZACIÓN ESPACIAL

6.4 MATRIZ DE RIGIDEZ

$$\int_0^L \left(D(x) \cdot \frac{\partial^2 u(x,t)}{\partial x^2} \cdot \frac{\partial^2 v(x)}{\partial x^2} \right) \cdot dx \quad \Rightarrow \quad \int_0^L \left(D(x) \cdot \frac{\partial^2 \hat{u}(x,t)}{\partial x^2} \cdot \frac{\partial^2 v(x)}{\partial x^2} \right) \cdot \mathbf{dx}$$

$$\int_0^L \left(D(x) \cdot \frac{\partial^2 \hat{u}(x,t)}{\partial x^2} \cdot \frac{\partial^2 v(x)}{\partial x^2} \right) \cdot \mathbf{dx} = \sum_e \int_e \left(D(x) \cdot \frac{\partial^2 \hat{u}(x,t)}{\partial x^2} \cdot \frac{\partial^2 v(x)}{\partial x^2} \right) \cdot \mathbf{dx}$$

$$\frac{\partial^2 \{\hat{u}^e(x,t)\}}{\partial x^2} = \frac{4}{L^2} \cdot \frac{\partial^2 \{\hat{u}^e(\xi,t)\}}{\partial \xi^2} = \frac{4}{L^2} \cdot \frac{\partial^2 \{N^e(\xi)\}}{\partial \xi^2} \cdot \{u^e(t)\} = \{B\} \cdot \{u^e(t)\} \quad \Rightarrow \quad \{B\} = \frac{1}{L} \cdot \left[6 \cdot \frac{\xi}{L} \quad 3\xi - 1 \quad -6\frac{\xi}{L} \quad 3\xi + 1 \right]$$

$$K^e = \int_{-1}^{1} D^e \cdot \{B\}^T \cdot \{B\} \cdot \frac{1}{2} \cdot L \cdot d\xi \quad \Rightarrow \quad [K^e] = D^e \begin{bmatrix} \frac{12}{L^3} & \frac{6}{L^2} & \frac{-12}{L^3} & \frac{6}{L^2} \\ \frac{6}{L^2} & \frac{4}{L} & \frac{-6}{L^2} & \frac{2}{L} \\ \frac{-12}{L^3} & \frac{-6}{L^2} & \frac{12}{L^3} & \frac{-6}{L^2} \\ \frac{6}{L^2} & \frac{2}{L} & \frac{-6}{L^2} & \frac{4}{L} \end{bmatrix}$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

6. DISCRETIZACIÓN ESPACIAL

6.5 MATRIZ DE MASA

$$\int_0^L \left(m(x)\cdot\frac{\partial^2\hat{u}}{\partial t^2}\right)\cdot \mathbf{v}\cdot \mathbf{dx} \quad \Rightarrow \quad [M^e]=m^e\cdot\int_{-1}^{1}\frac{L}{2}\cdot\{N^e\}^T\cdot\{N^e\}\cdot d\xi$$

$$[M^e]=\frac{m^e\cdot L}{420}\cdot\begin{bmatrix}156 & 22\cdot L & 54 & -13\cdot L\\ 22\cdot L & 4\cdot L^2 & 13\cdot L & -3\cdot L^2\\ 54 & 13\cdot L & 156 & -22\cdot L\\ -13\cdot L & -3\cdot L^2 & -22\cdot L & 4\cdot L^2\end{bmatrix}$$

- Matriz de masa concentrada (fácilmente invertible)
- $0 < \alpha < 1/50$

$$[M^e]=m^e\cdot L\cdot\begin{bmatrix}\frac{1}{2} & 0 & 0 & 0\\ 0 & \alpha\cdot L^2 & 0 & 0\\ 0 & 0 & \frac{1}{2} & 0\\ 0 & 0 & 0 & \alpha\cdot L^2\end{bmatrix}$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

6. DISCRETIZACIÓN ESPACIAL

6.6 VECTOR DE FUERZA

$$\int_0^L f(x,t)\cdot v\cdot dx \quad \Rightarrow \quad \{f^e\}=\int\{N(x)\}^T\cdot q^e(x,t)\cdot dx=\int_{-1}^{1}\{N(\xi)\}^T\cdot q^e(\xi,t)\cdot\frac{1}{2}\cdot L\cdot d\xi$$

$$\{f^e\}=\frac{1}{2}\cdot q^e\cdot L\cdot\int_{-1}^{1}\{N\}^T\cdot d\xi=\frac{1}{2}\cdot q^e\cdot L\cdot\int_{-1}^{1}\begin{bmatrix}\frac{1}{4}\cdot(\xi-1)^2\cdot(\xi+2)\\ \frac{1}{8}\cdot L^e\cdot(\xi+1)\cdot(\xi-1)^2\\ -\frac{1}{4}\cdot(\xi+1)^2\cdot(\xi-2)\\ \frac{1}{8}\cdot L^e\cdot(\xi+1)^2\cdot(\xi-1)\end{bmatrix}\cdot d\xi=\{q^e\}\cdot L\cdot\begin{bmatrix}\frac{1}{2}\\ \frac{L}{12}\\ \frac{1}{2}\\ -\frac{L}{12}\end{bmatrix}$$

- Posiciones impares, fuerzas aplicadas en los nodos.
- Posiciones pares, momentos aplicados en los nodos correspondiente.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

7. ANÁLISIS MODAL

- Partiendo de la ecuación semidiscreta,

$$[K]\cdot\{u\}+[M]\cdot\{\ddot{u}\}=0$$

y considerando el movimiento vibratorio de la forma:

$$\{u\}=\{U\}\cdot\sin(\omega\cdot t)$$

- A partir de lo anterior y sabiendo que [M] es regular:

$$[K]\cdot\{U\}=\omega^2\cdot[M]\cdot\{U\} \qquad [M]^{-1}\cdot[K]\cdot\{U\}=\omega^2\cdot\{U\}$$

- La frecuencia natural serán las raíces de los valores propios de $[M]^{-1}[K]$.

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

8. DISCRETIZACIÓN TEMPORAL

- Runge-Kutta_Nyström de orden 3.
- Condiciones iniciales:
  - U_0
  - V_0

$\frac{9+\sqrt{33}}{12}$	$\frac{(9+\sqrt{33})^2}{288}$	0
$\frac{1+\sqrt{33}}{6+2\sqrt{33}}$	$-\frac{99+19\sqrt{33}}{4(3+\sqrt{33})^2}$	$\frac{(9+\sqrt{33})^2}{288}$
	$\frac{3-\sqrt{33}}{54+6\sqrt{33}}$	$\frac{2(6+\sqrt{33})}{3(9+\sqrt{33})}$
	$\frac{2}{9+\sqrt{33}}$	$\frac{7+\sqrt{33}}{9+\sqrt{33}}$

$$Q_{n,i} = U_n + c_i \cdot k \cdot V_n + k^2 \cdot \sum_{i=1}^{s} a_{ij} \cdot F\left(t_n + c_j \cdot k, Q_{n,j}\right), \qquad i = 1, \dots, s.$$

$$V_{n+1} = V_n + k \cdot \textstyle\sum_{i=1}^{s} b_i \cdot F\left(t_n + c_i \cdot k, Q_{n,i}\right),$$

$$U_{n+1} = U_n + k \cdot V_n + k^2 \cdot \sum_{i=1}^{s} \beta_i \cdot F\left(t_n + c_i \cdot k, Q_{n,i}\right),$$

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

9. ALGUNOS EJEMPLOS

- Algunos ejemplos
  - Nueva carpeta\KCWMC_video.wmv
  - Nueva carpeta\rts.wmv
  - Nueva carpeta\Udmattelsestests.mpe

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira